ON THE SCORES AND THE ISOMORPHISM OF THE TOURNAMENTS

BY

MERAJ UDDIN

DEPARTMENT OF MATHEMATICS
INDIAN INSTITUTE OF TECHNOLOGY KANPUR

OCTOBER, 1983

ON THE SCORES AND THE ISOMORPHISM OF THE TOURNAMENTS

A Thesis Submitted
In Partial Fulfilment of the Requirements
for the Degree of
DOCTOR OF PHILOSOPHY

BY

MERAJ UDDIN

to the

DEPARTMENT OF MATHEMATICS
INDIAN INSTITUTE OF TECHNOLOGY KANPUR
OCTOBER, 1983

18/10/83
BR

CERTIFICATE

This is to certify that the Ph.D. thesis entitled 'On the Scores and the Isomorphism of the Tournaments' by Mr. Meraj Uddin is a record of bonafide research work carried out by him under my supervision and guidance. He had fulfilled the other requirements for the award of Ph.D. degree. The results embodied in this thesis have not been submitted to any other Institute or University for the award of any degree or diploma.

I.I.T.,Kanpur
October,1983

(M.R. Sridharan)
Professor of Mathematics
Indian Institute of Technology
Kanpur

ACKNOWLEDGMENTS

I take this opportunity to express my deep sense of gratitude and indebtedness to my thesis supervisor Professor M.R. Sridharan. Dr. M.R. Sridharan introduced me to graph theory and suggested many interesting research topics and provided excellent supervision, invaluable constructive criticism and constant encouragement at all stages of the research work. For these and his many other contributions, I thank him.

I would also like to thank all my family members and friends for the help which I received from them time to time. My special thanks are due to my wife for her patience and support.

I am grateful to Council of Scientific and Industrial Research, New Delhi, India, for the financial assistance for carrying out this research work.

My sincere thanks are to Mr. G.L. Mishra and Mr. S.K. Tiwari for typing and Mr. B.N. Shukla for cyclostyling.

Meraj Uddin

MERAJ UDDIN

CONTENTS

CHAPTER 1

INTRODUCTION

This chapter is divided into three sections. The first section deals with the history of the material related to our work presented in this thesis. In the second section we present the synopsis of the thesis and the last section consists of the basic definitions and notations.

1.1 BRIEF HISTORY There are many interesting areas of research in directed graphs. One such in the field of directed graphs is the tournaments. The tournament theory is one of the richest theories in directed graphs. There is no analogue of the tournament theory in the case of undirected graphs. The theory of tournaments is being studied for last many years but the combinatorial structure of the tournaments was studied only a few decades ago. Up to 1965, much research work was carried out on the tournaments but the results were all scattered. Harary et al [53] in 1965 for the first time made a systematic study of the tournaments and presented the known results in their book [53]. After a year only Harary et al [54] published a paper containing some more new results along with a bibliography The pioneering work on tournaments has been reported by Moon [82 in his book 'Topics on Tournaments', which was published in 1968. After 1968 much work has been reported on tournaments in different journals. A very useful and extensive survey as well

a bibliography (containing 95 references) of the work on tournaments has been published by Reid and Beineke [88] in 1978. This work of Reid and Beineke is less involved than Moon's [82]. Reid and Beineke have emphasised on recent work.

The important research areas of tournaments are the spanning paths, circuits, scores, enumeration, extremal problems, the automorphism of tournaments, regularity in tournaments, a variety of structural problems, isomorphism problems and many others. For the survey and bibliographies we refer to [88] and [107].

The literature on spanning paths and circuits can be found in [2,20,22,37,38,41,42,45,50,51,54,60,66,82,83,87,90,91,96,105].

The scores of tournaments,we study in detail in Chapter 2. For enumeration problems related with tournaments, we refer to [21,34,37,38,46,55,82,83] . The other references on tournaments can be found in [78,88,97,102,103,104,106,107,108].

It is interesting to see whether a part of the tournament theory can be applied to the other areas. The bipartite tournaments are the bipartite analogues of tournaments. Moon [80 for the first time studied the bipartite tournaments in detail in his doctoral dissertation in 1962. In the same year Moon and Moser [84] published a paper which deals with the distributio of 4-cycles in random bipartite tournaments. Beineke and Moon [10] have studied the score lists of bipartite tournaments, simpl

pair of bipartite score lists, the decomposition of bipartite tournaments into strong components and pair of score lists belonging to consistent bipartite tournaments. Beineke and Little [11] have studied the existence of cycles in bipartite tournaments. Beineke [9] has examined an analogous of the results on ordinary and bipartite tournaments. Beineke in this paper has studied bipartite tournaments in detail and has suggested some unsolved problems for further work. Bollabas et al [14] had determined recently the expected value and variance of the number of 4-cycles in mxn-bipartite tournaments and the probabili that an mxn-bipartite tournament has no cycles. Other references related to bipartite tournament can be found in [9].

The research work on tripartite tournaments is still in its early stage.

One of the unsolved challenging problems in computational graph theory is the graph isomorphism problem. More than 400 papers have appeared in the literature on graph isomorphism and related problems but graph isomorphism problem is not yet completely settled. The graph isomorphism problem is to devise an efficient algorithm for testing the isomorphism of a given pai of graphs. No polynomial time algorithm has yet been found i.e. the graph isomorphism problem is not known to be in P. For details we refer to [5,24,30,31,33,79,86]. Graph isomorphism problem is in NP. But unlike many other interactable problems it is not known to be NP-complete [24,31,33,47,49,69,75,79,86].

To have an overview of the work done on graph isomorphism and its present status we refer to[23,24,31,49,79,100] and the references given therein. Colbourn [23] has classified the graph isomorphism problem in many different areas. There are 3 main areas. First is the theoretical aspect of graph isomorphism problem. Second is the algorithmic aspect of the problem. The third is the surveys of the work on graph isomorphism. Each section is further divided in to subsections. For details we refer to [23]. Due to the unsettled nature of the problem the isomorphism of restricted families of graphs and other structures have been studied. There are certain families of graphs, structures whose isomorphism problem is polynomially time equivalent to graph isomorphism. Such problems are called the isomorphism complete problems. There are many known isomorphism complete problems of restricted families of graphs such as bipartite graphs [101], Chordal graphs and transitively orientable graphs [18], rooted acyclic digraphs [1 regular graphs [16,33,79], regular self-complementary graphs [26], k-trees for arbitrary k [65]. For many other isomorphism complete problems of families of graphs, we refer to [19]. There are many structures whose isomorphism problem is isomorphism complete such as semigroups, finite automata [17], lattices [44] finitely presented algebras [67]. We have noted that all these isomorphism complete problems were basically isomorphism problems However there are problems which are not isomorphism problems but

are isomorphism complete. Among such problems one is the k-clique problem [57,68,73].

There are certain families of graphs whose isomorphism problems are not known to be isomorphism complete. Among such problems there are certain families of graphs for which isomorphism testing is an easier task i.e. a polynomial time algorithm has been obtained. Some of the families of graphs whose isomorphism can be tested in polynomial time are, trees [58], Planar graphs [59], graphs with distinct eigenvalues [5], k-trees for k constant [56], interval graphs [76], transitive series parallel digraphs [71], cographs [72,95], permutation graphs [25], graphs with bounded valence [77]. Still there are certain families of graphs which are neither known to be isomorphism complete nor a polynomial time algorithm has been obtained. Tournament is one such family and we study tournament isomorphism problem in Chapter 3. For other such problems we refer to [19,33].

The continued interest in the research work on graph isomorphism problem is due to its complexity and many of its practical applications. For applications we refer to [24,86]. Many other references, not mentioned in this thesis, on the computational complexity of graph isomorphism problem can be found in [24,31,33,49,75,79]. For papers on probabilistic and random methods, expected time complexity and theoretical results on isomorphism of random graphs, we refer to [5,6,7,63, 64,74].

Lastly we give references of general graph isomorphism algorithms. The list consists of [12,13,28,29,30,32,35,36,61, 86,94,98,99,100].

1.2 SYNOPSIS OF THE THESIS. We give here a synopsis in order to mention the contributions contained in this thesis. Chapter 1 provides the necessary ground work to understand the work presented in the subsequent sections.

We study in Chapter 2, tournaments and their score sequences. Our main emphasis is on different kinds of score sequences as simple score sequences, self-converse score sequences, self-converse and simple score sequences. A recurrence relation is given to evaluate the number of self-converse simple score sequences of order n. The concepts of nearly-simple and near-simple score sequences have been introduced and are studied in detail. Some recurrence relations to evaluate nearly-simple, near-simple, self-converse nearly-simple and near-simple score sequences have been obtained.

We deal with the tournament isomorphism problem in Chapter 3. Tournament isomorphism problem is neither solvable in polynomial time nor is known to be isomorphism complete. We refer to Colbourn et al [26] for other details. We have established that the tournament isomorphism problem and the strong tournament isomorphism problem are polynomially equivalent problems. We also present an algorithm to test whether the two given tournaments are isomorphic or not.

Chapter 4 is on another class of tournaments known as bipartite tournaments. First we obtain some results on the bipartite score lists and the strong components of bipartite tournaments. A technique is given to obtain all the bipartite score lists of order mxn with the help of computer. The problem of bipartite tournament isomorphism is studied in the next section and some results are obtained. We have counted the number of strong simple pairs of bipartite score lists of order mxn and then give a recurrence relation to evaluate the total number of simple pairs of bipartite score lists of order mxn. In the last portion partial characterisations of self-converse bipartite score lists have been achieved

In the last Chapter, we discuss one more class of tournaments, known as the tripartite tournaments. First we characterise tripartite score-lists and present a technique to generate all the tripartite score lists of order pxqxr with the help of computer. In the end we study the relationship between the tripartite and the bipartite tournaments. The concept of score vectors is introduced and then the tripartite score vectors are characterised. Some analogous results of bipartite tournaments in the case of tripartite tournaments are obtained and some new results are reported. A few conjectures and some open problems are mentioned.

1.3 BASIC DEFINITIONS AND NOTATIONS

We introduce here definitions and concepts directly related to the tournaments and the graph isomorphism. Other important

definitions are standard and can be found in any text book of graph theory e.g. [8,15,35,43,52,53,55,82,88,89] . For definitions and notations on concepts of computational complexity of graph isomorphism, we refer to [1,27,40,47,48,62].

Some of the definitions are given below.

<u>Definition 1.3.1.</u> A graph G consists of a finite nonempty set V = V(G) of p vertices (points) known as vertex set, together with a prescribed set E of q unordered pairs of distinct vertices (points) of V. The set E is known as the edge set and the elements of E are known as edges (lines). If e = (a,b) is an edge, then e is said to join a and b.

<u>Definition 1.3.2.</u> A digraph D consists of a finite non-empty set V of nodes (points), known as the vertex set, together with a prescribed collection E of ordered pairs of distinct nodes, known as the arc set. The digraph D = (V,E) represents a digraph with node set V and arc set E.

If e = (u,v) is an arc of a digraph then a is adjacent to b and b is adjacent from a.

<u>Definition 1.3.3.</u> An oriented graph is a digraph having no symmetric pair of directed lines (arcs).

<u>Definition 1.3.4.</u> The out-degree od(v) of a vertex v is the number of vertices adjacent from it, and the in-degree id(v) is the number adjacent to it.

Definition 1.3.5. A walk in a digraph is an alternating sequence of vertices and arcs $v_1, e_1, v_2, \ldots, e_{n-1}, v_n$ in which $e_i = (v_i, v_{i+1})$. The length of such a walk is n, the number of occurrences of arcs in it. A closed walk has the same first and last vertices and a spanning walk contains all the vertices.

Definition 1.3.6. A path is a walk in which all the vertices are distinct.

Definition 1.3.7. A cycle is a closed path.

Definition 1.3.8. If there is a path from a vertex u to the vertex v, then v is said to be reachable from u.

Definition 1.3.9. A digraph is strong if every two vertices are mutually reachable.

Definition 1.3.10. A subgraph of a graph G is a graph having all of its vertices and edges in G.

Definition 1.3.11. For any subset S of V(G) of a graph G, the induced subgraph $\langle S \rangle$ is the maximal subgraph of G with vertex set S.

Definition 1.3.12. The converse digraph D' of D has the same vertex set as D and $(u,v) \in E(D')$ iff $(v,u) \in E(D)$.

Definition 1.3.13. The complete graph K_n has every pair of its n nodes adjacent.

<u>Definition 1.3.14.</u> A complete bipartite graph G is a graph whose vertex set V is partitioned into two disjoint non-empty subsets V_1 and V_2 such that (u,v) is an edge for each u in V_1 and v in V_2. If $|V_1| = m$ and $|V_2| = n$, we write $G = K_{m,n}$.

<u>Definition 1.3.15.</u> A complete tripartite graph G is a graph whose vertex set V is partitioned into three disjoint non-empty subsets V_1, V_2 and V_3 such that (u,v) is an ~~arc~~ edge for each u in V_i and each v in V_j for $i \neq j$. If $|V_1| = p$, $|V_2| = q$ and $|V_3| = r$, then we write $G = K_{p,q,r}$.

<u>Definition 1.3.16.</u> A strong component of a digraph is a maximal strong subgraph.

<u>Definition 1.3.17.</u> The condensation D^* of a digraph D has the strong components of D as the vertices of D^*, with an arc from a strong component S_i to another strong component S_j whenever there is atleast one arc in D from a vertex of S_i to a vertex of S_j.

<u>Definition 1.3.18.</u> Two graphs $G_1 = (V_1,E_1)$ and $G_2 = (V_2,E_2)$ are isomorphic, denoted by $G_1 \cong G_2$ if there exists a one to one correspondence between their vertex sets V_1 and V_2 which preserves adjacencies (Similar definition can be given for digraphs.)

<u>Definition 1.3.19.</u> A digraph D is self-converse if $D \cong D'$.

<u>Definition 1.3.20.</u> A tournament T = (V,E) is a complete oriented graph with vertex set V and arc set E i.e. for every pair of vertices u and v either (u,v) is an arc or (v,u) is an arc, but not both. This tournament is called an ordinary tournament.

Definition 1.3.21. A bipartite tournament T is a complete oriented bipartite graph. Thus vertex set V(T) is partitioned in to two disjoint non-empty sets X and Y, known as **the partite sets** such that two vertices are joined by an arc iff they lie in different partite sets.

Definition 1.3.22. A tripartite tournament T is a complete oriented tripartite graph. Thus the vertex set V(T) is partitioned into three disjoint non-empty sets X, Y and Z, known as the partite sets, such that the two vertices are joined by an arc iff they lie in different partite sets.

Definition 1.3.23. In any tournament a vertex u dominates v if (u,v) is an arc.

Definition 1.3.24. In a tournament, the score of a vertex v, denoted by s(v), is the number of vertices dominated by v. Thus s(v) is the out-degree of v.

The other graph theoretical definitions,not given here, can be found in the references given at the beginning of this section. Now we give some definitions related to the computational complexity of the graph isomorphism problem.

Definition 1.3.25. A problem is said to be in P if it can be solved in polynomial time on a one-tape deterministic **Turing** machine. Informally, this means that the problem can be solved in polynomial time on an ordinary computer.

<u>Definition 1.3.26</u>. A problem is in NP if it can be solved in polynomial time on a one tape non-deterministic Turing machine. For details we refer to Aho et al [1] and Even [40].

<u>Definition 1.3.27</u>. A problem P_1 is polynomially reducible to a problem P_2, denoted by $P_1 \; \alpha_p \; P_2$, if the existence of a polynomial algorithm for P_2 implies the existence of a polynomial algorithm for P_1. If $P_1 \; \alpha_p \; P_2$ and $P_2 \; \alpha_p \; P_1$, then P_1 and P_2 are said to be polynomially equivalent problems.

For NP-completeness, there are three definitions available in the literature, one by Karp [62], one by Cook [27] and another by Aho el al [1]. Here we give the most general definition given by Aho et al.

<u>Definition 1.3.28.</u> A problem in NP is called NP-complete if it has the following property : if the problem belongs to P then all problems in NP also belongs to P.

There are many interactable problems which are NP-complete, for example, subgraph isomorphism, Hamiltonian cycle, k-clique and many others. For NP-complete problems we refer to Garey, Johnson and Stockmeyer [47], Karp [62], Aho et al [1].

<u>Definition 1.3.29</u>. By a list $(x_1, x_2, \ldots, x_n)$ we mean a set of n-nonnegative integers in a nondecreasing order.

The end of a proof is denoted by ||.

CHAPTER 2

SCORE SEQUENCES OF TOURNAMENTS

In this chapter we study some properties of the score sequences of tournaments. Our main emphasis is on strong, simple, self-converse, self-converse and simple score sequences. The concepts of near-simple and nearly-simple score sequences have been introduced and some results are obtained. Recurrence relations to evaluate the number of self-converse and simple score sequences, near-simple score sequences, nearly-simple score sequences, self-converse and near-simple and nearly-simple score sequences of order n are reported.

2.1 SCORE SEQUENCES OF TOURNAMENTS AND THEIR STRONG COMPONENTS

Let $T = (V, E)$ be a tournament with vertex set $V = \{v_1, v_2, \ldots, v_n\}$. If $|V| = n$, then we call the tournament an n-tournament and the order of the tournament is n. The score structure or score sequence, denoted by S, of a tournament T is formed by listing the scores of the vertices of T in a nondecreasing order. Clearly a score sequence of an n-tournament is a set of n nonnegative integers not exceeding $n-1$. There are $(2n-1)_{c_{n-1}}$ vectors or lists $(x_1, x_2, \ldots, x_n)$ of nonnegative integers in nondecreasing order with $x_i \leq n-1$

for $1 \leq i \leq n$. But every list is not a score sequence as $(0,1,1)$ is not a score sequence. Though $(0,1,2)$ is a score sequence. We denote by $S = (s_1, s_2, \ldots, s_n)$ the score sequence of an n-tournament T.

<u>Definition 2.1.1</u>. A score sequence $S = (s_1, s_2, \ldots, s_n)$ is said to be realizable by a tournament T if there exists a tournament T with $V(T) = \{v_1, v_2, \ldots, v_n\}$ such that $s_i = s(v_i)$ for $1 \leq i \leq n$. Such a T is known as realisation of S. A score sequence $S = (s_1, s_2, \ldots, s_n)$ is called a score sequence of order n. A score sequence S is regular if $s_1 = s_2 = \ldots = s_n$ and near-regular if the maximum difference between its scores is one. A natural question is, ''what collection of n-nonnegative integers, not exceeding n-1, forms a score sequence?''. This question was settled by Landau [70]. Landau's proof included a reduction step which also provides an algorithm for getting a tournament with the given score list.

<u>Theorem 2.1.1</u> [70]. Let S be a nondecreasing list of n nonnegative integers not exceeding n-1, and let $S = (s_1, s_2, \ldots, s_n)$. Let S_1 be obtained from S by deleting one entry s_i and reducing $n-1-s_i$ largest entries of S by 1. Then S is a score sequence iff S_1 is.

We explain the reduction technique with the help of an example given below. The tournament obtained in this way is known as a canonical tournament and is denoted by $T^*(S)$.

<u>Example 2.1.1</u>. Let $S = (1,2,2,2,4,4)$. Here in each step we delete the final i.e. the last entry and maintain the nondecreasing property while reducing entries. We get the following.

$S = (1,2,2,2,4,4)$

$S_1 = (1,2,2,2,3)$

$S_2 = (1,1,2,2)$

$S_3 = (1,1,1)$

$S_4 = (0,1)$

$S_5 = (0)$

The canonical tournament $T^*(S)$ is described by the following adjacency relation.

$$T^*(S) : v_1(v_3);\ v_2(v_1,v_5);\ v_3(v_2,v_4);$$
$$v_4(v_1,v_2);\ v_5(v_1,v_3,v_4,v_6);$$
$$v_6(v_1,v_2,v_3,v_4)$$

In Theorem 2.1.1 it is a must to reduce the largest entries of S. Otherwise S_1 need not be a score sequence. For if $S = (1,1,1,4,4,4)$ and S_1 be obtained from S by deleting the last entry and reducing the first entry by 1, then $S_1 = (0,1,1,4,4)$ which is not a score sequence.

Now we state Landau's existence criterion. This is a nonconstructive criterion and it simply shows whether a given set of n nonnegative integers not exceeding n-1 forms a score sequence or not.

Theorem 2.1.2 [70]. A nondecreasing list of n nonnegative integers $(s_1, s_2, \ldots, s_n)$ is the score sequence of a tournament iff for $1 \leq k \leq n$

$$\sum_{i=1}^{k} s_i \geq \frac{1}{2} k (k-1), \text{ with equality for } k=n \qquad (2.1.1)$$

Definition 2.1.2. A score sequence S is said to be strong if all the tournaments T with score sequence S are strong. Theorem 2.1.2 induces a result to find which score sequences are strong.

Corollary 2.1.1 [70]. A nondecreasing list of nonnegative integers $(s_1, s_2, \ldots, s_n)$ is a strong score sequence iff for $1 \leq k \leq n-1$

$$\sum_{i=1}^{k} s_i > \frac{1}{2} k (k-1) \text{ and } \sum_{i=1}^{n} s_i = \frac{1}{2} n (n-1) \qquad (2.1.2)$$

Beineke and Eggleton (unpublished) have independently shown that in applying Theorem 2.1.2 and Corollary 2.1.1, one needs only to check the inequality in equations (2.1.1) and (2.1.2) for those values of k for which $s_k < s_{k+1}$ (and of course the final inequality).

Let T be a tournament with score sequence S. The strong components of S are the score sequences of the strong component of T. The following result gives all the strong components of T. This result is an extension of Theorem 11.13 [53]. For further details we refer to [4].

Theorem 2.1.3 [4]. Let T be an n-tournament and $S = (s_1, s_2, \ldots, s_n)$ be the score sequence of T. Suppose

$$\sum_{i=1}^{p} s_i = \frac{1}{2}p.(p-1) \tag{2.1.3}$$

$$\sum_{i=1}^{q} s_i = \frac{1}{2}q.(q-1) \tag{2.1.4}$$

and $$\sum_{i=1}^{k} s_i > \frac{1}{2}k.(k-1), \text{ for } p+1 \leq k \leq q-1 \tag{2.1.5}$$

where $0 \leq p < q \leq n$.

Then the subtournament induced by the vertices $\{v_{p+1}, \ldots, v_q\}$ is a strong component of T with score sequence $(s_{p+1}-p, \ldots, s_q-p)$.

The above theorem shows that the strong components of S are determined by the successive values of k for which

$$\sum_{i=1}^{k} s_i = \frac{1}{2} k(k-1), \quad 1 \leq k \leq n \tag{2.1.6}$$

As an illustration we consider the following example.

Example 2.1.2. Let $S = (1,1,2,2,5,5,6,6,9,9,9)$. We note that the equation (2.1.6) is satisfied for $k=4,8,11$. Thus the strong components of S are, in ascending order, $(1,1,2,2)$, $(1,1,2,2)$ and $(1,1,1)$.

Definition 2.1.3. Let $S_1 = (s_{11}, s_{12}, \ldots, s_{1m})$ and $S_2 = (s_{21}, s_{22}, \ldots, s_{2n})$ be two score sequences of order m and n respectively. We define

$S_1 + S_2 = (s_{11}, s_{12}, \ldots, s_{1m}, m+s_{21}, m+s_{22}, \ldots, m+s_{2n})$.

Clearly S_1+S_2 is a score sequence of order $m+n$. The plus operation defined above is not commutative but it is associative.

Let $S = (s_1, s_2, \ldots, s_n)$ be a score sequence of order n and let $S_1, S_2, \ldots, S_k$ be the strong components of S obtained from Theorem 2.1.3. The strong components $S_1, S_2, \ldots, S_k$ can be arranged in an ordered sequence $S_1, S_2, \ldots, S_k$ such that $S = S_1 + S_2 + \ldots + S_k$.

Such a decomposition is known as the strong component decomposition of S.

<u>Definition 2.1.4</u>. Let $T_1, T_2, \ldots, T_k$ be the tournaments with disjoint vertex sets. Then $T = [T_1, T_2, \ldots, T_k]$ denote the tournament obtained from the T_i, $1 \leq i \leq k$, by joining every vertex of T_j to all vertices of T_i with $1 \leq i < j \leq k$.

We know that the condensation of any tournament T is transitive order [53, pp. 297-298], [82, p. 2]. Hence strong components $T_1, T_2, \ldots, T_k$ of a tournament T can be arranged in an ordered sequence $T_1, T_2, \ldots, T_k$ such that $T = [T_1, T_2, \ldots, T_k]$.

Such decomposition is known as the strong component decomposition of T.

We observe that there are $(2n-1)_{C_{n-1}}$ lists of non-negative integers not exceeding n-1, but all of them are not score sequences. The number t(n) of different score sequences of order n can be obtained by a recursive technique of Narayna and Bent [85]. Also we refer to Moon [82, p. 67]. But no explicit formula has yet been reported. Below we list for some values of n, the number of score sequences of order n.

n =	1	2	3	4	5	6	7	8	9	10
t(n) =	1	1	2	4	9	22	59	167	490	1486

Table 2.1.1

We arrange all the t(n) score sequences of order n in a particular manner so that every score sequence is assigned a unique label or a unique position by which it will be recognised. This arrangement is explained below.

Definition 2.1.5. Let $X = (x_1, x_2, \ldots, x_n)$ and $Y = (y_1, y_2, \ldots, y_n)$ be two different nondecreasing lists of nonnegative integers. We say that X precedes Y, denoted by $X <_p Y$, iff $x_k < y_k$ and $x_i = y_i$ for $i = 1, 2, \ldots, k-1$ for some k, $1 \leq k \leq n$.

Example 2.1.3. Let $X = (1,1,1,3,4)$ and $Y = (1,1,2,2,4)$. In this case $X <_p Y$.

A list X is the immediate predecessor of a list Y if there exists no $Z = (z_1, z_2, \ldots, z_n)$ such that $X <_p Z < <_p Y$. Such an ordering of lists is known as antilexicographic

ordering. We arrange all the $t(n)$ score sequences of order n in antilexicographic order and assign labels $1,2,\ldots,t(n)$ to the score sequences. Such a collection is denoted by $\tau(n)$. Example 2.1.4 represents all the 9 score sequences of order 5 arranged in antilexicographic order.

Example 2.1.4. Let $O(T) = 5$

Sequence Number	Score Sequence
1	(0,1,2,3,4)
2	(0,1,3,3,3)
3	(0,2,2,2,4)
4	(0,2,2,3,3)
5	(1,1,1,3,4)
6	(1,1,2,2,4)
7	(1,1,2,3,3)
8	(1,2,2,2,3)
9	(2,2,2,2,2)

The third score sequence of order 5 is (0,2,2,2,4) while the 7th score sequence of order 5 is (1,1,2,3,3). Let $S = (s_1,s_2,\ldots,s_n)$ be the score sequence of a tournament T. Then $S' = (n-1-s_n,\ldots,n-1-s_1)$ is the score sequence of T', the converse of T.

Now we establish some results.

Theorem 2.1.4. Let $S_1 = (s_1,s_2,\ldots,s_n)$ and $S_2 = (0, s_1+1,\ldots,s_n+1)$. S_1 is the mth score sequence of $\tau(n)$ iff S_2 is the mth score sequence of $\tau(n+1)$.

Proof. Let T_1 be a realisation of S_1. Then $T_2 = [K, T_1]$, where K is a tournament of order 1, is a realisation of S_2. This shows that S_2 is a score sequence when S_1 is a score sequence. Let T be a realisation of S_2. We can write $T = [T_1, T_2]$, where T_1 is a tournament of order 1. Clearly T_2 is a realisation of S_1. This shows that S_1 is a score sequence when S_2 is a score sequence. The unique correspondence shows that both are occupying the same position.||

Let $t_k(n)$ denote the number of score sequences of order n having score k at least once, for $0 \leq k \leq n-1$. The following results are of much interest.

Theorem 2.1.5. $t_k(n) = t_{n-1-k}(n)$, for $0 \leq k \leq n-1$.

Proof. This is equivalent to proving that whenever $S = (s_1, s_2, \ldots, s_n)$ is a score sequence then $S' = (n-1-s_n, \ldots, n-1-s_1)$ is also a score sequence. But this always happens since S is the score sequence of a tournament T iff S' is the score sequence of tournament T'.||

Theorem 2.1.6. $t_o(n) = t(n-1)$.

Proof 1. Let $S_1 = (s_1, s_2, \ldots, s_{n-1})$ be the last i.e. $t(n-1)^{th}$ score sequence of order $n-1$. By Theorem 2.1.4, $S_2 = (0, s_1+1, \ldots, s_{n-1}+1)$ is the $t(n-1)^{th}$ score sequence of order n. Now we show that there does not exist any score sequence $S_3 = (t_1, t_2, \ldots, t_n)$, $S_3 \neq S_2$ such that $t_1 = 0$ and $S_2 <_p S_3$. Suppose that there exists one S_3. Then by

Theorem 2.1.4, $S_4 = (t_2-1,\ldots,t_n-1)$ is a score sequence of order $n-1$ and $S_1 <_p S_4$. But this is not possible as S_1 is the last score sequence of $\tau(n-1)$. Thus S_2 is the last score sequence of $\tau(n)$ in which the first entry is zero. Hence $t_o(n) = t(n-1)$.||

Proof. 2. Let $S = (s_1, s_2, \ldots, s_n)$ be a score sequence of order n with the strong component decomposition $S = S_1 + S_2 + \ldots + S_k$. We are interested in those score sequences S of order n such that $S_1 = (0)$. Deletion of the entry corresponding to this trivial component leaves the order of the score sequence as $n-1$. Thus there are $t(n-1)$ score sequences of order n for which $S_1 = (0)$. Hence $t_o(n) = t(n-1)$.||

Corollary 2.1.2. $t_{n-1}(n) = t(n-1)$.

Proof. By Theorem 2.1.5, $t_{n-1}(n) = t_o(n)$ and hence by the previc result $t_{n-1}(n) = t(n-1)$.||

The significance of Theorem 2.1.6 and Corollary 2.1.2 is that there are $t(n-1)$ score sequences out of $t(n)$ score sequences in $\tau(n)$ having either receivers or transmitters.

Let $\tau_{rt}(n)$ denote the number of score sequences of order n, which are having receivers and transmitters both. We prove the following result.

Theorem 2.1.7. $t_{rt}(n) = t(n-2)$

Proof 1. In $\tau(n-1)$ out of $t(n-1)$ score sequences we have

$t(n-2)$ score sequences which are having transmitters by corollary 2.1.2. In $\tau(n)$ first $t(n-1)$ score sequences are having receivers by Theorem 2.1.6. Out of these $t(n-1)$ score sequences having receivers, only $t(n-2)$ score sequences are having transmitters. Thus $t_{rt}(n) = t(n-2)$.||

Proof 2. Let $S = (s_1, s_2, \ldots, s_n)$ be a score sequence of order n with the strong component decomposition $S = S_1 + \ldots + S_k$. We are interested in those score sequences S of order n such that $S_1 = (0)$ and $S_k = (0)$. Deletion of these entries leaves the order of the score sequences as $n-2$. Thus there are $t(n-2)$ score sequences of order n such that $S_1 = S_k = (0)$. Hence $t_{rt}(n) = t(n-2)$.||

Thus in the light of the above result we observe that there are only $t(n-2)$ score sequences of order n, which have receivers and transmitters both.

Let $tt(n)$ denote the number of strong score sequences of order n. A strong nontrivial tournament contains atleast three vertices and so a nontrivial strong score sequence must have order atleast three. No result is known which gives the values of $tt(n)$ for all $n \geq 3$. Table 2.1.2 lists the values of $tt(n)$ for some values of n. This may help to get a formula to evaluate the values of $tt(n)$ for different values of n.

n	=	3	4	5	6	7	8	9	10	11	12
tt(n)	=	1	1	3	7	21	61	184	573	1835	5969

Table 2.1.2

2.2 SIMPLE SCORE SEQUENCES

A score sequence $S = (s_1, s_2, \ldots, s_n)$ is said to be simple if all the tournaments having this score sequence are isomorphic to each other. Thus a simple score sequence belongs to exactly one tournament. All the score sequences of order upto 4 are simple. (1,1,2,3,3) and (1,2,2,2,3) are the only two score sequences of order 5 which are not simple. We know that there are two non-isomorphic tournaments with the score sequence (1,1,2,3,3) and three nonisomorphic tournaments with the score sequence (1,2,2,2,3). Avery [4] has characterised the simple score sequences.

Lemma 2.2.1 [4]. A score sequence S is simple iff every strong component of S is simple.

This lemma shows that one has to only characterise the simple strong score sequences. This characterisation is obtained by the following result.

Theorem 2.2.1 [4] A strong score sequence S is simple iff it is one of (0), (1,1,1), (1,1,2,2) and (2,2,2,2,2).

Combining lemma 2.2.1 and Theorem 2.2.1, we observe that a score sequence S is simple iff each of its strong components is one of (0), (1,1,1), (1,1,2,2) and (2,2,2,2,2).

Example 2.2.1. Let $S = (1,1,1,4,4,5,5,7,10,10,10,10,10)$. The strong components of S are (1,1,1), (1,1,2,2), (0) and

$(2,2,2,2,2)$. Therefore S is simple. But $S=(1,1,1,4,4,5,6,6)$ is not simple as one of its strong components $(1,1,2,3,3)$ is not simple.

Let $s(n)$ denote the number of simple score sequences of order n. Avery [4] gave a recurrence relation to evaluate $s(n)$. Table 2.2.1 lists for some values of n, the number of simple score sequences of order n.

n	=	1	2	3	4	5	6	7	8	9	10
s(n)	=	1	1	2	4	7	11	18	31	53	89

Table 2.2.1

Now we establish some results of simple score sequences.

<u>Theorem 2.2.2</u>. A score sequence S is simple iff S' is simple.

<u>Proof</u>. Let S be simple and suppose that S' is not simple. Let T_1 and T_2 be the two nonisomorphic tournaments with score sequence S'. This shows that $T_1' \not\cong T_2'$. But S is the score sequence of T_1' and T_2'. This contradicts the fact that S is simple. Hence our assumption that S' is not simple is wrong. Similarly the converse can also be established.∎

Let $s_k(n)$ for $0 \leq k \leq n-1$, denote the number of simple score sequences of order n having score k atleast once. The following results are of much interest.

<u>Theorem 2.2.3</u>. Let $S_1 = (s_1,s_2,\ldots,s_n)$ and $S_2 = (0,s_1+1,\ldots,s_n+1)$. S_1 is simple iff S_2 is simple.

Proof 1. The strong components of S_2, in ascending order, are (0) and the strong components of S_1. Thus S_2 is simple iff S_1 is simple.||

Proof 2. Let T_1 be the realisation of the simple score sequence S_1. Let $T_2 = [K_1, T_1]$ where K_1 is a tournament of order 1. Clearly T_2 is the realisation of S_2. Hence S_2 is simple. Similarly converse can also be established.||

We have seen in Theorem 2.1.4 that $S_1 = (s_1, s_2, \ldots, s_n)$ is the m^{th} score sequence in $\tau(n)$ iff $S_2 = (0, s_1+1, \ldots, s_n+1)$ is the m^{th} score sequence in $\tau(n+1)$. Thus by Theorem 2.2.3 if the m^{th} score sequence of order n is simple then all the m^{th} score sequences of order greater than n are also simple. Thus we can say that the positions of simple score sequences are well behaved.

Theorem 2.2.4. $s_k(n) = s_{n-1-k}(n)$ for $0 \leq k \leq n-1$.

Proof. This result is equivalent to showing that if S is simple then S' is also simple. This follows from Theorem 2.2.2.||

Theorem 2.2.5. $s_0(n) = s(n-1)$.

Proof 1. Let $S = (s_1, s_2, \ldots, s_{n-1})$ be the last i.e. $s(n-1)$th simple score sequence of order $n-1$ (i.e. there does not exist any other simple score sequence S_1 of order $n-1$ such that $S <_p S_1$ and $S \neq S_1$). By Theorem 2.2.3, $S_2 = (0, s_1+1, \ldots, s_n+1)$ is also simple and its position is

$s(n-1)$th in $\tau(n)$ by Theorem 2.1.4. Now we show that there does not exist any other simple score sequence $S_3 = (x_1, \ldots, x_n)$ such that $x_1 = 0$ and $S_2 <_p S_3$ and $S_2 \neq S_3$. On the contrary if such an S_3 exists, then $S_4 = (x_2-1, \ldots, x_n-1)$ is a simple score sequence in $\tau(n-1)$ and $S <_p S_4$. This is a contradiction to the fact that S is the last simple score sequence in $\tau(n-1)$. Hence S_2 is the last simple score sequence in $\tau(n)$ such that its first entry is zero. So $s_o(n) = s(n-1)$.||

Proof 2. Let $S = (s_1, s_2, \ldots, s_n)$ be a simple score sequence of order n with the strong component decomposition $S = S_1 + \ldots + S_k$. Clearly each S_i is simple. We are interested in those simple score sequences S of order n for which $S_1 = (0)$. Deletion of this entry reduces the order of the score sequence to $n-1$. Thus there are $s(n-1)$ simple score sequences of order n such that $S_1 = (0)$. Hence $s_o(n) = s(n-1)$.||

Corollary 2.2.1. $s_{n-1}(n) = s(n-1)$.

Proof. $s_{n-1}(n) = s_o(n)$ by Theorem 2.2.4 but $s_o(n) = s(n-1)$ by Theorem 2.2.5. Hence the result.||

Therefore there are $s(n-1)$ simple score sequences of order n which are having either receivers or transmitters.

Let $s_{rt}(n)$ denote the number of simple score sequences of order n having receivers and transmitters both. Now we have the following result.

<u>Theorem 2.2.6.</u> $s_{rt}(n) = s(n-2)$.

<u>Proof 1.</u> In $\tau(n-1)$ there are $s(n-2)$ simple score sequences which are having transmitters by Corollary 2.2.1. In $\tau(n)$ the first $t(n-1)$ score sequences are having receivers by Theorem 2.1.6. Out of these $t(n-1)$ score sequences in $\tau(n)$ having receivers, $s(n-1)$ are simple. Thus there are $s(n-2)$ simple score sequences of order n having receivers and transmitters both. Therefore $s_{rt}(n) = s(n-2)$.||

<u>Proof 2.</u> Let $S = (s_1, s_2, \ldots, s_n)$ be a simple score sequence of order n with the strong component decomposition $S = S_1 + \ldots + S_k$. Each S_i is simple. We are interested in those simple score sequences S of order n such that $S_1=(0)$ and $S_k = (0)$. Deletion of the entries corresponding to these trivial components leaves the order of the score sequence as $n-2$. Thus there are $s(n-2)$ simple score sequences of order n such that $S_1 = S_k = (0)$. Hence $S_{rt}(n)=s(n-2)$.||

This result exhibits that there are only $s(n-2)$ simple score sequences of order n which are having receivers and transmitters both.

We know that a simple score sequence belongs to exactly one tournament. A natural question is, ''what is the number of nonisomorphic tournaments having a score sequence S which is not simple?'' Let $S = S_1 + S_2 + \ldots + S_k$ be the strong

component decomposition of S. Let n_i be the number of nonisomorphic strong tournaments with the strong score sequence S_i for $i = 1,2,\ldots,k$. The following result partially answers the question raised above. Let N be the number of nonisomorphic tournaments with the score sequence S. We have the following result.

<u>Theorem 2.2.7</u>. $N = \prod_{i=1}^{k} n_i$.

<u>Proof</u>. The proof is by mathematical induction on k. If $k = 1$ then the result is true. If S is simple then each of its strong components is also simple i.e. $n_i = 1$ for $1 \leq i \leq k$. In this case $N = 1$ and hence the result is true. Let the result be true for $k = m-1$. Let $S_1,\ldots,S_{m-1},S_m$ be the strong components of S. Each tournament with the strong components $S_1,S_2,\ldots,S_{m-1}$ is combined with n_m tournaments with score sequence S_m. Thus the number of tournaments with strong components $S_1,\ldots,S_{m-1},S_m$ equals $\prod_{i=1}^{m} n_i$.

Below,we state some remarks which can be easily established by using computer. The programme and the out puts are available with the author.

<u>Remark 2.2.1</u>. Consider $\tau(n)$. The score sequences 1 to 6 for $n \geq 5$; 9 to 12 and 20^{th} for $n \geq 6$; 23 to 28 and 49^{th} for $n \geq 7$; 60 to 65, 68 to 72 and 130 to 131 for $n \geq 8$ and so on, are simple.

Remark 2.2.2. Consider $\tau(n+1)$. The score sequences $t(n)+1$ to $t(n)+6$ for $n \geq 6$; $t(n)+9$ to $t(n)+12$ for $n \geq 7$; $t(n) + 20^{th}$ and $t(n) + 23$ to $t(n) + 28$ for $n \geq 8$ are simple.

Remark 2.2.3. There are no three consecutive simple score sequences in $\tau(n)$. That is to say if m-1th, mth and m+1th are three consecutive score sequences in $\tau(n)$, then they are not simple unless m-2th or m+2th is simple.

Remark 2.2.4. If m-2, m-1, m, m+1, m+2th are the only five consecutive simple score sequences (i.e. m-3rd and m+3rd score sequences are not simple), then m+62th and m+63rd are the only two consecutive simple score sequences (i.e. m+61th and m+64th score sequences are not simple) which are the immediate successors of the given simple score sequences.

We observe that the score sequences 68 to 72th of order 8 are simple and the immediate successors are the simple score sequences 130 and 131.

Remark 2.2.5. If m and m+1 are the only two consecutive simple score sequences in $\tau(n)$, then m+38 to m+43 are the simple score sequences which are the immediate successors.

Now we state a conjecture

Conjecture 2.2.1. For $0 \leq k \leq n-1$,

$$s_k(n) = s_k(n-1) + s_k(n-3) + s_k(n-4) + s_k(n-5) \qquad (2.2.1)$$

where $s_k(n) = 0$ if $k \geq n$

$s_k(k+1) = s(k)$, $s_1(3) = 2$, $s_1(4) = 3$, $s_2(5) = 5$.

We observe that the conjecture is true for $k = 0$ as $s_o(n) = s(n-1)$ by Theorem 2.2.5. By Corollary 2.2.1, the conjecture is true for $k = n-1$. This gives a motivation that this conjecture seems to be true. However a general proof has not yet been obtained.

2.3 SELF-CONVERSE SCORE SEQUENCES

A score sequence $S = (s_1, s_2, \ldots, s_n)$ is said to be self-converse if all the tournaments T having the score sequence S are self-converse i.e. $T \cong T'$. If $S=(s_1, s_2, \ldots, s_n)$ is the score sequence of a tournament T, then S', the score sequence of T', is $S' = (n-1-s_n, \ldots, n-1-s_1)$. We note that all the score sequences of order fewer than four are self-converse. The score sequence $(0,2,2,2)$ of order four is not self-converse. We also observe that all the strong score sequences upto order 5 are self-converse. $S=(1,1,3,3,3,4)$ is a strong score sequence of order six but is not self-converse.

Eplett [39] has characterised the self-converse score sequences.

__Theorem 2.3.1__ [39]. A score sequence $S = (s_1, s_2, \ldots, s_n)$ is self-converse iff

$$s_i + s_{n+1-i} = n-1 \quad \text{for} \quad 1 \leq i \leq n \tag{2.3.1}$$

This theorem is equivalent to saying that S is self-converse iff $S = S'$. One needs to verify equation (2.3.1) only for $i = 1,2,\ldots,[\frac{n}{2}]$. Every regular and near-regular score sequences are self-converse.

Let S be a score sequence and $S = S_1 + S_2 + \ldots + S_k$ be its strong component decomposition. Then $S' = S'_k + S'_{k-1} + \ldots + S'_1$ will be the strong component decomposition of S'. Thus S is self-converse iff

$$S_i = S'_{k+1-i} \quad \text{for} \quad 1 \leq i \leq k \tag{2.3.2}$$

We had observed that all the strong score sequences of order upto 5 are self-converse. Thus upto order 11, if a score sequence is self-converse, then each of its strong components is also self-converse. But for order 12, we have $S = (1,1,3,3,3,4,7,8,8,8,10,10)$ which is self-converse but its strong components $(1,1,3,3,3,4)$ and $(1,2,2,2,4,4)$ are not self-converse. All the strong components of a score sequence S may be self-converse but it does not guarentee that S is also self-converse. We take $S = (0,2,2,2)$, both the strong components (0) and $(1,1,1)$ are self-converse, but S is not self-converse.

Now we present techniques which give rise to self-converse score sequences.

<u>Theorem 2.3.2</u>. Let $S = (s_1, s_2, \ldots, s_n)$ be a score sequence. Then $S+S'$ is a self-converse score sequence.

Proof 1. We get $S' = (n-1-s_n,\ldots,n-1-s_1)$. Hence

$$S + S' = (s_1,s_2,\ldots,s_n,\ 2n-1-s_n,\ldots,2n-1-s_2,\ 2n-1-s_1)$$

$$= (x_1,\ x_2,\ldots,x_{2n}) .$$

where $x_i = s_i$ for $1 \leq i \leq n$ and

$x_i = 2n-1-s_{2n+1-i}$ for $n+1 \leq i \leq 2n$

We note that $x_i + x_{2n+1-i} = 2n-1$ for $1 \leq i \leq 2n$.

Thus by Theorem 2.3.1, $S + S'$ is self-converse.||

Proof 2. Let $S = S_1 + S_2 + \ldots + S_k$ be the strong component decomposition of S. Then $S' = S_k' + \ldots + S_2' + S_1'$. Thus

$$S + S' = S_1 + S_2 + \ldots + S_k + S_k' + \ldots + S_2' + S_1'$$

Therefore

$$(S+S')' = (S_1+S_2+\ldots+S_k+S_k'+\ldots+S_2'+S_1')'$$

$$= S_1+S_2+\ldots+S_k+S_k'+\ldots+S_2'+S_1'.$$

As $(S_i')' = S_i$ for $1 \leq i \leq k$. So we get

$(S + S')' = S + S'$, Hence $S + S'$ is self-converse.||

Theorem 2.3.3. Let $S = (s_1,s_2,\ldots,s_n)$ be a self-converse score sequence and S_1 be any other score sequence. Then S_1+S+S_1' is a self-converse score sequence.

Proof 1. Let $S_1 = (t_1,\ t_2,\ldots,\ t_m)$. Then

$S_1' = (m-1-t_m,\ldots,m-1-t_2,m-1-t_1)$. Thus

$$S_1 + S + S_1' = (t_1,t_2,\ldots,t_m,\ m+s_1,\ldots,m+s_n,$$
$$2m+n-1-t_m,\ldots,2m+n-1-t_1) \qquad (2.3.3)$$

$$= (x_1,\ldots,x_m,\ x_{m+1},\ldots,x_{m+n},\ x_{m+n+1},\ldots,x_{2m+n}) \qquad (2.3.4)$$

where

$$\begin{aligned} x_i &= t_i \quad \text{for } 1 \leq i \leq m, \\ x_{m+i} &= m+s_i \quad \text{for } 1 \leq i \leq n, \\ x_{m+n+i} &= 2m+n-1-t_{m+1-i} \quad \text{for } 1 \leq i \leq m \end{aligned} \qquad (2.3.5)$$

Now $x_j + x_{2m+n+1-j} = x_j + x_{m+n+(m+1-j)}$

$= t_j+2m+n-1-t_{m+1-(m+1-j)}$ for $1 \leq j \leq m$

(from equation (2.3.5))

$$= 2m+n-1 \quad \text{for } 1 \leq j \leq m \qquad (2.3.6)$$

Also $x_j + x_{2m+n+1-j} = x_{m+(j-m)} + x_{m+(m+n+1-j)}$

$= m + s_{j-m} + m + s_{m+n+1-j}$ for $m+1 \leq j \leq m+n$

$= 2m + s_{j-m} + s_{n+1-(j-m)}$

$= 2m+n-1$, as S is self-converse $\qquad (2.3.7)$

From equations (2.3.6) and (2.3.7), we get that

$$x_j + x_{2m+n+1-j} = 2m+n-1 \quad \text{for } 1 \leq j \leq 2m+n$$

Hence by Theorem 2.3.1, $S_1 + S + S_1'$, is self-converse.||

<u>Proof 2</u>. Let $S^* = S_1 + S + S_1'$. We note that

$$S^{*\prime} = (S_1 + S + S_1')' = (S_1')' + S' + S_1'$$

$$= S_1 + S' + S_1',$$

but S' is self-converse and hence $S' = S$. Therefore $S^{*\prime} = S_1 + S + S_1' = S^*$ i.e. S^* is self-converse.||

Let $ts(n)$ and $tts(n)$ denote the number of self-converse score sequences and the number of self-converse strong score sequences of order n respectively. Table 2.3.1 lists the values of $ts(n)$ for some values of n.

n =	1	2	3	4	5	6	7	8	9	10	11
ts(n) =	1	1	2	2	5	6	15	19	48	46	161

Table 2.3.1

An exact formula to evaluate $ts(n)$ has not yet been reported.

Table 2.3.2 lists the values of $tts(n)$ for some values of n.

n =	3	4	5	6	7	8	9	10	11
tts(n) =	1	1	3	3	9	11	30	39	103

Table 2.3.2

Let $ts_k(n)$, $0 \leq k \leq n-1$ and $tts_k(n)$, $1 \leq k \leq n-2$ denote the number of self-converse and self-converse strong score sequences respectively, of order n, having the score k atleast once.

The following results follow immediately from Theorem 2.3.1.

<u>Theorem 2.3.4.</u> $ts_k(n) = ts_{n-1-k}(n)$ for $0 \leq k \leq n-1$.

<u>Theorem 2.3.5.</u> $tts_k(n) = tts_{n-1-k}(n)$ for $1 \leq k \leq n-2$.

We conclude this section with the following result.

<u>Theorem 2.3.6.</u> $ts_o(n) = ts(n-2)$.

<u>Proof.</u> Let $S = (s_1, s_2, \ldots, s_n)$ be a self-converse score sequence of order n. Let the strong component decomposition of S be $S = S_1 + S_2 + \ldots + S_k$. Hence $S' = S_k' + \ldots + S_2' + S_1'$. S being self-converse, we have $S = S'$ iff $S_i' = S_{k+1-i}$ for $1 \leq i \leq k$. We are interested in those self-converse score sequences for which $S_1 = (0)$. If $S_1 = (0)$, then $S_1' = (0) = S_k$. The deletion of the vertices corresponding to the trivial components S_1 and S_k reduces the order of the tournament to n-2. Thus there are ts(n-2) self-converse score sequences such that $S_1 = (0)$. Therefore $ts_o(n) = ts(n-2)$.||

The following corollary follows from Theorems 2.3.4 and 2.3.6.

<u>Corollary 2.3.1.</u> $ts_{n-1}(n) = ts(n-2)$.

Theorem 2.3.6 and Corollary 2.3.1 show that there are ts(n-2) self-converse score sequences which are having receivers and transmitters.

2.4 SELF-CONVERSE SIMPLE SCORE SEQUENCES

A score sequence S which is both self-converse and simple will be called a self-converse simple score sequence. In this section, we study the self-converse simple score sequences. $S = (0,1,2,\ldots,n-1)$ is a self-converse simple score sequence. The following lemma can be easily established.

Lemma 2.4.1. A score sequence S is self-converse and simple iff it is self-converse and each of its strong components is simple.

Let $ss(n)$ denote the number of self-converse simple score sequences of order n, and $ss_k(n)$ for $0 \leq k \leq n-1$, denote the number of self-converse simple score sequences of order n having score k atleast once. Below, we establish a recurrence relation which can be used to evaluate $ss(n)$ for all values of n.

Theorem 2.4.1. $ss(n) = ss(n-2) + ss(n-6) + ss(n-8) + ss(n-10)$ (2.4.1)

where $ss(k) = 0$ if $k < 0$,

$ss(0) = ss(1) = ss(2) = 1$, $ss(3) = ss(4) = 2$ and $ss(5) = 3$.

Proof. By the lemma 2.4.1, S is self-converse simple iff S is self-converse and each strong component of S is simple. Let $h(k)$ denote the number of strong simple score sequences of order k. Let $S = S_1 + S_2 + \ldots + S_k$ be the strong component

decomposition of S. Then $S' = S'_k + \dots + S'_2 + S'_1$ is the strong component decomposition of S'. As S is self-converse, $S = S'$ i.e. $S_i = S'_{k+1-i}$ for $1 \leq i \leq k$. Hence

$$ss(n) = \sum_{k=1}^{[n/2]} h(k)\ ss(n-2k) \qquad (2.4.2)$$

where $ss(k) = 0$ if $k < 0$, $ss(0) = ss(1) = ss(2) = 1$, $ss(3) = ss(4) = 2$ and $ss(5) = 3$. But (0), (1,1,1), (1,1,2,2) and (2,2,2,2,2) are the only strong simple score sequences by Theorem 2.2.1. Thus $h(1) = h(3) = h(4) = h(5) = 1$ and $h(k) = 0$ for other values of k. Substituting these values of $h(k)$ in equation (2.4.2), we get equation (2.4.1) which establishes the result.||

Table 2.4.1 lists $ss(n)$ for some values of n.

n =	1	2	3	4	5	6	7	8	9	10	11
ss(n)=	1	1	2	2	3	3	4	5	7	9	13

Table 2.4.1

The self-converse simple score sequences can be generated by the following results.

<u>Theorem 2.4.2</u>. If S is simple, then $S+S'$ is a self-converse simple score sequence.

<u>Proof</u>. $S+S'$ is self-converse by Theorem 2.3.2. Since S is simple, S' is simple by the Theorem 2.2.2. The strong components of the score sequence $S+S'$ are the strong components of S and S'. But S and S' being simple,

all the strong components of S and S' are simple. Thus each of the strong components of $S+S'$ is also simple and hence $S+S'$ is simple by lemma 2.2.1. Therefore $S+S'$ is self-converse and simple.||

<u>Theorem 2.4.3</u>. Let S be a self-converse simple score sequence and SA be any other simple score sequence. Then $SA + S+SA'$ is a self-converse simple score sequence.

<u>Proof</u>. $SA+S+SA'$ is self-converse by Theorem 2.3.3 . SA being simple, SA' is also simple by Theorem 2.2.2. The strong components of $SA+S+SA'$ are the strong components of SA, S and SA'. As SA, S and SA' are all simple score sequences, the strong components of SA, S and SA' are all simple by lemma 2.2.1. Hence each of the strong components of $SA+S+SA'$ is also simple. Therefore $SA+S+SA'$ is simple by lemma 2.2.1. Thus $SA+S+SA'$ is a self-converse simple score sequence.||

The following result is of some interest.

<u>Theorem 2.4.4</u>. $ss_k(n) = ss_{n-1-k}(n)$ for $0 \leq k \leq n-1$.

<u>Proof</u>. It is equivalent to showing that whenever $S = (s_1, s_2, \ldots, s_n)$ is a self-converse simple score sequence, so is S'. S' is simple by Theorem 2.2.2. S' is self-converse as $(S')' = S = S'$. Therefore, S' is self-converse simple.||

<u>Theorem 2.4.5</u>. $ss_0(n) = ss(n-2)$.

Proof. Let $S = (s_1, s_2, \ldots, s_n)$ be a self-converse and simple score sequence of order n. Clearly S' is also self-converse and simple. Let $S = S_1 + \ldots + S_k$ be the strong component decomposition of S. S being simple, each S_i is simple. S being self-converse i.e. $S' = S$ and hence $S_i' = S_{k+1-i}$ for $1 \leq i \leq k$, where $S' = S_k' + \ldots + S_2' + S_1'$. We are interested in those self-converse and simple score sequences S of order n such that $S_1 = (0)$. Hence $S_k = S_1' = (0)$. By deleting the vertices corresponding to the trivial components S_1 and S_k the order of S is n-2. Thus there are ss(n-2) self-converse and simple score sequences of order n such that $S_1 = (0)$. Hence $ss_o(n) = ss(n-2)$.||

Following is an immediate consequence of Theorem 2.4.4 and 2.4.5.

Corollary 2.4.1. $ss_{n-1}(n) = ss(n-2)$.

Thus there are ss(n-2) self-converse simple score sequences of order n which are having receivers and transmitters both.

Remark 2.4.1. The first score sequence of $\tau(n)$ is always self-converse simple for $n \geq 1$.

Remark 2.4.2. In $\tau(n+1)$, the t(n)+2nd score sequence for $n \geq 5$ is always self-converse simple.

Remark 2.4.3. In $\tau(n+2)$, the t(n) + 3rd score sequence is always self-converse simple for $n \geq 6$.

2.5 NEARLY-SIMPLE AND NEAR-SIMPLE SCORE SEQUENCES.

Let $S = (s_1, s_2, \ldots, s_n)$ be a score sequence of order n. Let SR be obtained from S by deleting one entry s_i and reducing $n-1-s_i$ largest entries of S by 1. Theorem 2.1.1 guarentees that SR is also a score sequence. We give some definitions.

Definition 2.5.1. A strong score sequence S is nearly-simple if SR, obtained from S by deleting a particular entry of S, called a root, is simple. A score sequence S is said to be nearly-simple iff each of its strong components is nearly-simple.

Definition 2.5.2. A strong score sequence S is near-simple if SR, obtained from S by deleting any entry of S, is simple. A score sequence S is said to be near-simple iff each of its strong components is near-simple.

Clearly every simple score sequence is nearly-simple as well as near-simple. Every near-simple score sequence is nearly-simple but the converse need not be true.

In this section first we characterise nearly-(near) simple strong score sequences which are not simple and then we establish some more results.

Theorem 2.5.1. A strong score sequence S is nearly-simple (but not simple) iff S is one of (1,1,2,3,3), (1,2,2,2,3), (2,2,2,3,3,3) (any entry can be deleted), (1,2,3,3,3,3)

(only first entry can be deleted), (2,2,2,2,3,4) (only last entry can be deleted).

Proof. Let S be (1,1,2,3,3) or (1,2,2,2,3). Then SR, obtained by deleting any entry, is (1,1,2,2) which is simple. Let S be (1,2,3,3,3,3). Then SR, obtained by deleting the first entry, is (2,2,2,2,2) which is simple. Let S be (2,2,2,2,3,4) Then SR, obtained by deleting the last entry, is (2,2,2,2,2) which is simple. If S is (2,2,2,3,3,3), then SR, obtained by deleting any entry, is (2,2,2,2,2) which is simple. This proves one part.

Conversely let SR be a strong simple score sequence. Then we have the following possibilities.

(I) SR is (0) or (1,1,1). In this case S has to be of order 2 and 4 respectively. But all the score sequences of order fewer than 5 are simple. Hence S has to be simple. Thus SR can be neither (0) nor (1,1,1).

(II) SR is (1,1,2,2). Then S is of order 5 and is strong but not simple. There are only two strong but not simple score sequences of order 5, namely (1,1,2,3,3) and (1,2,2,2,3). Deletion of any entry of (1,1,2,3,3) or (1,2,2,2,3) gives SR to be (1,1,2,2). Thus S can be (1,1,2,3,3) or (1,2,2,2,3).

(III) SR is (2,2,2,2,2). Then S is of order 6 and is strong but not simple. But the strong not simple score sequences of order 6 are (1,1,2,3,4,4), (1,1,3,3,3,4), (1,2,2,2,4,4),

(1,2,2,3,3,4), (1,2,3,3,3,3), (2,2,2,2,3,4) and (2,2,2,3,3,3). Only (1,2,3,3,3,3) (deletion of first entry), (2,2,2,2,3,4) (deletion of last entry) and (2,2,2,3,3,3) (deletion of any entry) give rise SR = (2,2,2,2,2). Thus S can be (1,2,3,3,3,3) or (2,2,2,2,2,3,4) or (2,2,2,3,3,3). This completes the proof.||

From the above result, we have

Corollary 2.5.1. A score sequence S is nearly-simple iff each of its strong components is one of (0), (1,1,1), (1,1,2,2), (1,1,2,3,3), (1,2,2,2,3), (2,2,2,2,2), (1,2,3,3,3,3), (2,2,2,2,3,4) or (2,2,2,3,3,3).

We can check with the help of the above result whether a score sequence is nearly-simple or not.

The proof of Theorem 2.5.1 leads to the following result.

Theorem 2.5.2. A strong score sequence S is near-simple (but not simple) iff S is one of (1,1,2,3,3), (1,2,2,2,3) or (2,2,2,3,3,3).

Thus we obtain a result which enables us to check whether the given score sequence is near-simple or not.

Corollary 2.5.2. A score sequence S is near-simple iff each of its strong components is one of (0), (1,1,1), (1,1,2,2), (1,1,2,3,3), (1,2,2,2,3), (2,2,2,2,2) or (2,2,2,3,3,3).

Let $sr(n)$ $(st(n))$ denote the number of nearly-simple (near-simple) score sequences of order n. We obtain some recurrence relations to evaluate the values of $sr(n)$ and $st(n)$ for all values of n.

<u>Theorem 2.5.3</u>. $sr(n) = sr(n-1) + sr(n-3) + sr(n-4)$
$$+ 3sr(n-5) + 3sr(n-6) \tag{2.5.1}$$
where $sr(k) = 0$ if $k < 0$ and $sr(0) = 1$.

<u>Proof</u>. Let $h(k)$ denote the number of strong nearly-simple score sequences of order k. A score sequence S is nearly-simple iff each of its strong component is nearly-simple.

Thus
$$sr(n) = \sum_{k=1}^{n} h(k)\, sr(n-k) \tag{2.5.2}$$
where $sr(k) = 0$ if $k < 0$ and $sr(0) = 1$.

By the corollary 2.5.1, the strong nearly-simple score sequences are (0), $(1,1,1)$, $(1,1,2,2)$, $(2,2,2,2,2)$, $(1,1,2,3,3)$, $(1,2,2,2,3)$, $(1,2,3,3,3,3)$, $(2,2,2,2,3,4)$ and $(2,2,2,3,3,3)$. Thus $h(1) = h(3) = h(4) = 1$, $h(5) = h(6) = 3$ and $h(k) = 0$ for all other values of k. Substituting these values in equation (2.5.2), we get equation (2.5.1). This proves the result.||

Now we present a recurrence relation which can be used to evaluate $st(n)$, the number of near-simple score sequences of order n. The proof is similar to the proof of Theorem 2.5.3 and hence we omit it.

<u>Theorem 2.5.4.</u> $st(n) = st(n-1)+st(n-3) + st(n-4) + 3st(n-5)$ (2.5.3)

where $st(k) = 0$ if $k < 0$ and $st(0) = 1$.

Table 2.5.1 lists the values of $sr(n)$ and $st(n)$ for some values of n.

n	=	1	2	3	4	5	6	7	8	9
sr(n)	=	1	1	2	4	9	18	30	52	97
st(n)	=	1	1	2	4	9	16	26	46	85

Table 2.5.1

Now, we establish some results for nearly-simple and near-simple score sequences.

<u>Theorem 2.5.5.</u> Let $S_1 = (s_1, s_2, \ldots, s_n)$ be the m^{th} nearly-simple (near-simple) score sequence of order n. Then all the m^{th} score sequences of order greater than n are nearly-simple (near-simple).

<u>Proof.</u> If $S_1 = (s_1, s_2, \ldots, s_n)$ is the m^{th} score sequence of order n, then $S_2 = (0, s_1+1, \ldots, s_n+1)$ is the m^{th} score sequence of order n+1 by the Theorem 2.1.4. The strong components of S_2 are $\{(0)$, strong components of $S_1\}$. Thus S_2 is a nearly-simple (near-simple) iff S_1 is nearly-simple (near-simple) and this establishes the result.||

The above result shows that like simple score sequences the positions of nearly-simple (near-simple) score sequences in $\tau(n)$ are also well behaved.

Now we prove a result which is an analogue of Theorem 2.2.2 but the technique of the proof is different.

Theorem 2.5.6. A score sequence S is nearly-simple iff S' is nearly-simple.

Proof. Let S be a nearly-simple score sequence and $S = S_1+S_2+\ldots+S_k$ be the strong component decomposition of S. Then $S' = S_k'+\ldots+S_2'+S_1'$ is the strong component decomposition of S'. S being nearly-simple each strong component S_i of S is one of (0), $(1,1,1)$, $(1,1,2,2)$, $(2,2,2,2,2)$, $(1,1,2,3,3)$, $(1,2,2,2,3)$, $(1,2,3,3,3,3)$, $(2,2,2,2,3,4)$ or $(2,2,2,3,3,3)$. Except $(1,2,3,3,3,3)$ and $(2,2,2,2,3,4)$, other nearly simple strong score sequences are self-converse. We observe that if $SA = (1,2,3,3,3,3)$, then $SA'=(2,2,2,2,3,4)$. Thus we note that the converse of a strong nearly-simple score sequence is also nearly-simple. Hence S is nearly-simple iff S' is.||

The above proof suggests one more proof of Theorem 2.2.2.

Similarly we can prove the following result.

Corollary 2.5.3. A score sequence S is near-simple iff S' is near-simple.

Let $sr_k(n)$, $0 \leq k \leq n-1$, denote the number of nearly-simple score sequences of order n having the score k atleast once.

Theorem 2.5.7. $sr_k(n) = sr_{n-1-k}(n)$, for $0 \leq k \leq n-1$.

Proof. This is equivalent to showing that whenever S is a nearly-simple score sequence then S' is also a nearly-simple score sequence. This follows from Theorem 2.5.6.||

Let $st_k(n)$ denote the number of near-simple score sequences of order n having score k atleast once for $0 \leq k \leq n-1$. Similar to Theorem 2.5.7 we obtain the following result.

Corollary 2.5.4. $st_k(n) = st_{n-1-k}(n)$ for $0 \leq k \leq n-1$.

Theorem 2.5.8. $sr_o(n) = sr(n-1)$ and $st_o(n) = st(n-1)$.

Proof. The proof is similar to that of Theorem 2.2.5.||

The following corollary can easily be proved.

Corollary 2.5.5. $sr_{n-1}(n) = sr(n-1)$ and $st_{n-1}(n) = st(n-1)$.

Thus there are $sr(n-1)$ nearly-simple and $st(n-1)$ near-simple score sequences of order n which are having receivers or transmitters.

Let $sr_{rt}(n)$ and $st_{rt}(n)$ denote the number of nearly-simple and near-simple score sequences of order n having both receivers and transmitters respectively. The following result can easily be established like Theorem 2.2.6.

Theorem 2.5.9. $sr_{rt}(n) = sr(n-2)$ and $st_{rt}(n) = st(n-2)$.

This shows that there are only $sr(n-2)$ nearly-simple and $st(n-2)$ near-simple score sequences of order n having receivers and transmitters both.

We know that a simple score sequence belongs to exactly one tournament. A nearly-simple and a near-simple score sequence may belong to more than one nonisomorphic tournaments. A natural question is, ''how many nonisomorphic tournaments are there with a near-simple or nearly-simple score sequence S?'' We answer this question partially here. We discuss the problem : how many nonisomorphic tournaments are there with a strong nearly-simple score sequence S, having $T^*(SR)$ as its subtournament where SR is the simple score sequence obtained from S by deleting one entry?

If S is one of (0), (1,1,1), (1,1,2,2) or (2,2,2,2,2), then there is only one tournament with score sequence S having $T^*(SR)$ as its subtournament. Regarding the other strong nearly-simple score sequences, we discuss them one by one.

(I) S is (1,1,2,3,3). Then SR = (1,1,2,2) obtained by deleting any entry of S. $T^*(SR)$ is as given below by its adjacencies. Here we are taking the vertex set to be $V=\{1,2,\ldots,n\}$

$T^*(SR)$: 1(3); 2(1); 3(2,4); 4(1,2)

The following are the only two nonisomorphic tournaments with score sequence S having $T^*(SR)$ as its subtournament.

T_1: 1(3); 2(1); 3(2,4); 4(1,2,5); 5(1,2,3)

T_2: 1(3); 2(1); 3(2,4,5); 4(1,2); 5(1,2,4).

(II) S is (1,2,2,2,3). Then SR = (1,1,2,2), obtained by deleting any entry of S. T^*(SR) is the same as in case (I) There are only three nonisomorphic tournaments with the score sequence S having T^*(SR) as its subtournament.

T_1 : 1(3); 2(1,5); 3(2,4); 4(1,2); 5(1,3,4)

T_2 : 1(3,5); 2(1); 3(2,4); 4(1,2); 5(2,3,4)

T_3 : 1(3); 2(1,5); 3(2,4,5); 4(1,2); 5(1,4)

(III) S is (1,2,3,3,3,3). Then SR = (2,2,2,2,2) obtained by deleting first entry of S. T^*(SR) is as given below.

T^*(SR) : 1(3,5); 2(1,5); 3(2,4); 4(1,2); 5(3,4)

There is only one tournament T_1 with the score sequence S having T^*(SR) as its subtournament.

T_1 : 1(3,5); 2(1,5,6); 3(2,4,6); 4(1,2,6); 5(3,4,6); 6(1).

This does not mean that there is only one tournament with the score sequence S. As

T_2 : 1(6); 2(1,3); 3(1,4,5); 4(1,2,5); 5(1,2,6); 6(2,3,4)

is a tournament with the score sequence S but T^*(SR) is not a subtournament of T_2 and $T_2 \not\cong T_1$.

(IV) S is (2,2,2,2,3,4). Then SR = (2,2,2,2,2) obtained by deleting the last entry of S. T^*(SR) is the same as

in the previous case(III). There is only one tournament T_1 with score sequence S having $T^*(SR)$ as its subtournament.

T_1 : 1(3,5); 2(1,5); 3(2,4); 4(1,2); 5(3,4,6); 6(1,2,3,4)

This does not mean that there is only one tournament T_1 with the score sequence S. As

T_2 : 1(2,6); 2(3,4); 3(1,4); 4(1,5); 5(1,2,3); 6(2,3,4,5)

is a tournament with score sequence S but $T^*(SR)$ is not a subtournament of T_2 and $T_2 \not\cong T_1$.

(IV) S is (2,2,2,3,3,3). Then SR = (2,2,2,2,2) obtained by deleting any entry of S. $T^*(SR)$ is same as in case(III). There are only three non-isomorphic tournaments with score sequence S having $T^*(SR)$ as its subtournament.

T_1 : 1(3,5); 2(1,5); 3(2,4); 4(1,2,6); 5(3,4,6); 6(1,2,3)

T_2 : 1(3,5); 2(1,5); 3(2,4,6); 4(1,2); 5(3,4,6); 6(1,2,4)

T_3 : 1(3,5); 2(1,5,6); 3(2,4); 4(1,2,6); 5(3,4,6); 6(1,3)

2.6 SELF-CONVERSE NEARLY-SIMPLE AND NEAR-SIMPLE SCORE SEQUENCES

A score sequence S is self-converse nearly-simple (near-simple) if it is self-converse and nearly-simple (near-simple). In this section, we study the self-converse nearly-simple and near-simple score sequences. Two recurrence relations to evaluate the number of self-converse nearly simple and near-simple score sequences of order n are obtained.

Let $ssr(n)$ $(sst(n))$ denote the number of self-converse nearly-simple (near-simple) score sequences of order n and suppose $ssr_k(n)$ $(sst_k(n))$ for $0 \leq k \leq n-1$, denote the number of self-converse nearly-simple (near-simple) score sequences of order n having the score k atleast once. First we establish a recurrence relation which can be used to evaluate $ssr(n)$.

Theorem 2.6.1. $ssr(n) = ssr(n-2) + ssr(n-6) + ssr(n-8)$
$$+ 3ssr(n-10) + 3ssr(n-12) \qquad (2.6.1)$$

where $ssr(k) = 0$ if $k < 0$, $ssr(0) = ssr(1) = ssr(2) = 1$, $ssr(3) = ssr(4) = 2$, $ssr(5) = 5$ and $ssr(6) = 4$.

Proof. Let S be a self-converse nearly-simple score sequence of order n and $S = S_1 + S_2 + \ldots + S_k$ be its strong component decomposition. Then $S' = S'_k + \ldots + S'_2 + S'_1$ is the strong component decomposition of S'. As S is self-converse, $S = S'$ and hence $S_i = S'_{k+1-i}$ for $1 \leq i \leq k$. Let $h(k)$ denote the number of strong nearly-simple score sequences of order k. Then

$$ssr(n) = \sum_{k=1}^{[n/2]} h(k)\ ssr(n-2k) \qquad (2.6.2)$$

where $ssr(k) = 0$ if $k < 0$,

$ssr(0) = ssr(1) = ssr(2) = 1$, $ssr(3) = ssr(4) = 2$, $ssr(5) = 5$, and $ssr(6) = 4$.

In the proof of Theorem 2.5.3 we noted that $h(1) = h(3) = h(4) = 1$, $h(5) = h(6) = 3$ and $h(k) = 0$ for all other values of k. Substituting these values of $h(k)$ in equation (2.6.2), we get the result. $\|$

The following result can be similarly established.

Theorem 2.6.2. $sst(n) = sst(n-2) + sst(n-6) + sst(n-8) + 3sst(n-10) + sst(n-12)$ (2.6.3)

where $sst(k) = 0$ if $k < 0$, $sst(0) = sst(1) = sst(2) = 1$, $sst(3) = sst(4) = 2$, $sst(5) = 5$, and $sst(6) = 4$.

This result can be used to evaluate $sst(n)$ for different values of n.

Table 2.6.1 lists $ssr(n)$ and $sst(n)$ for some values of n

n =	1	2	3	4	5	6	7	8	9	10	11	12	13
ssr(n) =	1	1	2	2	5	4	6	6	9	12	19	24	39
sst(n) =	1	1	2	2	5	4	6	6	9	12	19	22	37

Table 2.6.1

Self-converse nearly-simple score sequences can be generated with the help of the following results.

Theorem 2.6.3. If S is a nearly-simple score sequence, then $S+S'$ is a self-converse nearly-simple score sequence.

Proof. $S+S'$ is self-converse by Theorem 2.3.2. We have to only show that $S+S'$ is nearly simple. By Theorem 2.5.6 S' is nearly simple as S is nearly-simple. The strong components of $S+S'$, in ascending order are the strong components of S, and the strong components of S', in ascending order. But the strong components of

S and S' are nearly-simple. Hence the strong components of $S+S'$ are also nearly-simple and thus $S+S'$ is nearly-simple.||

Similarly the following result can also be established.

Theorem 2.6.4. If S is near-simple score sequence, then $S+S'$ is a self-converse near-simple score sequence.

Theorem 2.6.5. Let S be a self-converse nearly-simple score sequence and SA be any other nearly-simple score sequence. Then $SA+S+SA'$ is a self-converse nearly-simple score sequence.

Proof. $SA+S+SA'$ is self-converse by Theorem 2.3.2. Since SA is nearly-simple, SA' is nearly-simple by Theorem 2.5.6. Thus all the strong components of $SA+S+SA'$ which are the strong components of SA, in ascending order and the strong components of S, in ascending order, and the strong components of SA', in ascending order are nearly-simple. Hence $SA+S+SA'$ is nearly-simple.||

In the same way, we can prove the following result.

Theorem 2.6.6. Let S be a self-converse near-simple score sequence and SA be any other near-simple score sequence. Then $SA+S+SA'$ is a self-converse near-simple score sequence.

Theorem 2.6.7. $ssr_k(n) = ssr_{n-1-k}(n)$ for $0 \leq k \leq n-1$.

Proof. It is sufficient to show that whenever $S = (s_1, s_2, \ldots, s_n)$

is a self-converse nearly-simple score sequence, then S' is also a self converse nearly-simple score sequence. S' is nearly-simple from Theorem 2.5.6 and it is self-converse as $(S')' = S = S'$. This establishes the result.||

Similarly we can prove the following result.

<u>Theorem 2.6.8.</u> $sst_k(n) = sst_{n-1-k}(n)$ for $0 \leq k \leq n-1$.

We conclude this section with the following results.

<u>Theorem 2.6.9.</u> $ssr_0(n) = ssr(n-2)$ and $sst_0(n) = sst(n-2)$.

<u>Proof.</u> The proof is similar to that of Theorem 2.4.5.||

The following is an immediate consequence of Theorems 2.6.8 and 2.6.9.

<u>Corollary 2.6.1.</u> $ssr_{n-1}(n) = ssr(n-2)$ and $sst_{n-1}(n) = sst(n-2)$.

Thus we observe that there are $ssr(n-2)$ $(sst(n-2))$ nearly-simple (near-simple) score sequences of order n which are having receivers and transmitters.

CHAPTER 3

TOURNAMENT ISOMORPHISM

The graph isomorphism problem is to decide whether two given graphs are isomorphic or not. Many algorithms, based on graph invariants, have been reported e.g. [30,31,86]. But no algorithm which runs in polynomial time for graph isomorphism is known i.e. the problem is not known to be in P. Unlike many other combinatorial problems, graph isomorphism problem is not known to be NP-complete see [24,30,31,33,79,86]. Thus the computational complexity of the graph isomorphism problem is still open. Due to this nature of the graph isomorphism problem, isomorphism of restricted families of graphs has been studied. There are certain families of graphs for which isomorphism is an easier task i.e. polynomial time algorithms have been obtained e.g. [5,21,56,58,59,71,72,76,77,95]. But for certain restricted families of graphs, isomorphism is polynomially time equivalent to graph isomorphism i.e. the family of graphs is isomorphism complete e.g. [16,17,19,26,31,33,68]. But tournament isomorphism problem is neither isomorphism complete nor a polynomial time algorithm is known. In this chapter, we study the tournament isomorphism problem.

First we note that the tournament isomorphism problem is not isomorphism complete. We exhibit that the tournament isomorphism and near-regular tournament isomorphism are

polynomially time equivalent problems. We also show that the tournament isomorphism problem and strong tournament isomorphism problem are polynomially time equivalent. In the last we discuss tournament isomorphism algorithm.

3.1 ISOMORPHISM COMPLETENESS OF TOURNAMENTS. Most of the proofs of isomorphism completeness transform an arbitrary graph G to a graph H which uniquely represents G, where H is the member of that family of graphs whose isomorphism completeness has to be proved. There are many well known techniques available. For details, we refer to [19]. In all the reduction techniques the automorphism group of the resulting graph H contains the automorphism group of the input graph G as a subgroup i.e. automorphism group of a graph is preserved under the transformation of isomorphism completeness. No reduction technique is available which transforms an arbitrary graph to a tournament such that the resulting tournament uniquely represents the graph. We also know that the automorphism group of a graph may be isomorphic to a group of any order. But the order of the automorphism group of a tournament is always odd, see [81]. This suggests that there can not exist any transformation, computable in polynomial time of the size of the input graph, which transforms an arbitrary graph to a tournament. Thus the isomorphism problem of tournaments can not be isomorphism complete. Due to the structure of the tournaments and the fact that the isomorphism problem of tournaments is not isomorphism complete, it appears

that the isomorphism testing of tournament should be an easy task. This fact was earlier noted by Colbourn et al [26].

3.2 TOURNAMENT ISOMORPHISM PROBLEM

In Section 3.1, we have observed that the tournament isomorphism problem is not isomorphism complete. Colbourn et al [26] have shown that the isomorphism testing of tournaments and regular self-converse tournaments (every regular tournament is self-converse) are polynomially time equivalent problems. First, we show that isomorphism testing of tournaments and near-regular tournaments (a tournament is near-regular if the maximum difference between its scores is 1) are polynomially time equivalent problems.

Theorem 3.2.1. The isomorphism of tournaments and near-regular tournaments are polynomially time equivalent problems.

Proof. Let T be a tournament with vertex set $V(T) = \{v_1, v_2, \ldots, v_n\}$. We consider two copies T_1 and T_2 of T. Let $V(T_i) = \{v_{i1}, v_{i2}, \ldots \ldots, v_{in}\}$ for $i = 1,2$. We define a tournament $NR(T)$ as follows.

The vertex set $V(NR(T))$ of $NR(T)$ is the union of $V(T_1)$ and $V(T_2)$. The arc set $E(NR(T))$ of $NR(T)$ consists of the arcs of T_1 and T_2 and the following arcs.

(I) $(v_{1i}, v_{2j}) \in E(NR(T))$ iff $(v_j, v_i) \in E(T)$.

(II) $(v_{2j}, v_{1i}) \in E(NR(T))$ iff either $j = i$ or $(v_i, v_j) \in E(T)$.

Clearly $s(v_{1i}) = n-1$ for $1 \leq i \leq n$ and $s(v_{2j}) = n$ for $1 \leq j \leq n$. Thus $NR(T)$ is a near-regular tournament. $NR(T)$

uniquely represents T as the vertices with scores n-1 (or n) induces a copy of T. Thus isomorphism of tournaments and near-regular tournaments are polynomially time equivalent problem. ||

As every near-regular tournament is self-converse, so we have also established the following result.

Corollary 3.2.1. The isomorphism testing of tournaments and near-regular self-converse tournaments are polynomially time equivalent problems.

Now, we show that the isomorphism testing of tournaments and strong tournaments are polynomially time equivalent problems. Thus Colbourn et al's [26] result as well as Theorem 3.2.1 become a particular case of the following result as every regular and near-regular tournaments are strong.

Let T_1 and T_2 be two tournaments to be tested for isomorphism. Let T_1 and T_2 have the following strong component decompositions

$$T_1 = [T_{11}, T_{12}, \ldots, T_{1k}]$$

and

$$T_2 = [T_{21}, T_{22}, \ldots, T_{2m}]$$

We know that $T_1 \cong T_2$ iff $k = m$ and $T_{1i} \cong T_{2i}$ for $1 \leq i \leq k$.

Thus we have the following important result.

Theorem 3.2.2. The tournament isomorphism problem and the strong tournament isomorphism problem are polynomially time equivalent.

We observe that the proof of Theorem 3.2.2 is simply obtained with the help of the structural properties of tournaments. But Colbourn et al [26] in their proof have transformed a tournament T of order n to a regular tournament R(T), of order 4n+1, which uniquely represents T. In the proof of Theorem 3.2.1 we have transformed a tournament T of order n to a near-regular tournament NR(T) of order 2n, where NR(T) uniquely represents T.

3.3 TOURNAMENT ISOMORPHISM TECHNIQUE

In this section, we shall discuss the refinement procedure applicable to tournaments. In the light of Section 3.2, we will concentrate only on strong tournaments. There is plenty of literature available on graph isomorphism algorithms. For the details of graph isomorphism algorithms we refer to Colbourn [23] and Read and Corneil [86] and the references given therein. The technique of the tournament isomorphism which we shall discuss is based on refinement procedure given by Corneil and Gotlieb [32] and Corneil [31]. For the sake of completeness, we briefly present the refinement procedure here. But we shall be considering only tournaments not graphs as we are interested in the tournament isomorphism.

3.3.1 REFINEMENT PROCEDURE: Let T = (V,E) be a tournament and S be the score sequence of T. The refinement procedure is as follows.

Step 1. We partition the vertex set V on the basis of the score sequence. So if T is regular, then V can not be partitioned on the basis of the score sequence and hence we go to Step 6. If V has been partitioned, we go to Step 2.

Step 2. Let V be partitioned in to the disjoint cells $V_1, V_2, \ldots$ $\ldots, V_m$. Now to each vertex u of V, we associate a vector (or list) of dimension m such that its ith entry is the number of vertices in V_i which are being dominated by u. We go to Step 3.

Step 3. The vertices of each cell are re-arranged by arranging the vectors of each cell in antilexicographic order.

Step 4. Further refinement is done by partitioning the vertices of the cells which have got different lists on the basis of lists associated with vertices. We reindex each cell. If the number of cells equals the number of vertices, then we say that the final partition has been achieved and we stop. If any cell is further refined and final partition is not acheived, we go to Step 2. If no cell has been further refined we go to Step 5.

Step 5. Let V_i be the first cell such that $|V_i| > 1$. We choose a vertex of this cell and place it in the first cell and reindex all the ith cell to i+1th cell, we go to Step 2.

Step 6. If T is regular, we choose a vertex and place it in the first cell. We now place other vertices in the 2nd cell and go to Step 2.

When refinement procedure is applied on a tournament, we always get the final partition.

We shall now discuss different examples of tournaments which will explain the refinement procedure more clearly. All these examples have been worked on computer by using the refinement procedure program.

Example 3.3.1. This is an example of a tournament in which the refinement procedure gives final partition but Step 5 and Step 6 are not required.

Let the tournament T be given by the following adjacencies. Here we take $V(T) = \{1,2,\ldots,n\}$.

T : 1(3); 2(1); 3(2,6); 4(1,2,3,5); 5(1,2,3,6); 6(1,2,4,7);
7(1,2,3,4,5)

The score sequence S is given by

Vertex label L(I) =	1	2	3	4	5	6	7
Score sequence S(I) =	1	1	2	4	4	4	5

On the basis of the score sequence (applying Step 1) the vertex set is partitioned as follows.

Cell No.	Vertices
1	1 2
2	3
3	4 5 6
4	7

Now using Step 2, we get

Cell No.	Vertices	Lists
1	1 2	(0,1,0,0) (1,0,0,0)
2	3	(1,0,1,0)
3	4 5 6	(2,1,1,0) (2,1,1,0) (2,0,1,1)
4	7	(2,1,2,0).

The application of Step 3 gives the following (vectors of each cell are arranged in antilexicographic order).

Cell No.	Vertices	Lists
1	1 2	(0,1,0,0) (1,0,0,0)
2	3	(1,0,1,0)
3	6 5 4	(2,0,1,1) (2,1,1,0) (2,1,1,0)
4	7	(2,1,2,0).

The application of Step 4 further refines the Cells 1 and 3. (Now we go to Step 2 and arrange the lists in antilexicographic order).

Cell No.	Vertices	List
1	1	(0,0,1,0,0,0)
2	2	(1,0,0,0,0,0)
3	3	(0,1,0,1,0,0)
4	6	(1,1,0,0,1,1)
5	4 5	(1,1,1,0,1,0) (1,1,1,1,0,0)
6	7	(1,1,1,0,2,0)

Cell 5 is further refined and we observe that the final partition is achieved.

Cell No.	Vertices	Lists
1	1	(0,0,1,C,0,0,0)
2	2	(1,0,0,0,0,0,0)
3	3	(0,1,0,1,0,0,0)
4	6	(1,1,0,0,1,0,1)
5	4	(1,1,1,0,0,1,0)
6	5	(1,1,1,1,0,0,0)
7	7	(1,1,1,0,1,1,0)

We note that the number of iterations required to get the final partition is 3.

<u>Example 3.3.2</u>. This is an example of a tournament in which Step 5 is used to get the final partition. Let T be a tournament given by the following adjacency relation. Here we simply report the computer out put and do not explain all the steps (for details see, example 3.4.1).

T : 1(7); 2(1); 3(1,2); 4(1,2,3,5); 5(1,2,3,6); 6(1,2,3,4); 7(2,3,4,5,6)

Vertex label L(I) = 1 2 3 4 5 6 7

Score sequence S(I)=1 1 2 4 4 4 5

Iteration No. = 1

Cell No.	Vertices	Lists (arranged in antilexicographic order)
1	1 2	(0,0,0,1) (1,0,0,0)
2	3	(2,0,0,0)
3	4 5 6	(2,1,1,0) (2,1,1,0) (2,1,1,0)
4	7	(1,1,3,0)

Iteration No. = 2

Cell No.	Vertices	Lists
1	1	(0,0,0,0,1)
2	2	(1,0,0,0,0)
3	3	(1,1,0,0,0)
4	4 5 6	(1,1,1,1,0) (1,1,1,1,0) (1,1,1,1,0)
5	7	(0,1,1,3,0)

No further partitioning, on the basis of lists associated with vertices, is possible and hence we apply Step 5.

Iteration No. = 3

Cell No.	Vertices	Lists
1	4	(0,1,1,1,1,0)
2	1	(0,0,0,0,0,1)
3	2	(0,1,0,0,0,0)
4	3	(0,1,1,0,0,0)
5	5 6	(0,1,1,1,1,0) (1,1,1,1,0,0)
6	7	(1,0,1,1,2,0)

Iteration No. = 4.

Cell No.	Vertices	Lists
1	4	(0,1,1,1,1,0,0)
2	1	(0,0,0,0,0,0,1)
3	2	(0,1,0,0,0,0,0)
4	3	(0,1,1,0,0,0,0)
5	5	(0,1,1,1,0,1,0)
6	6	(1,1,1,1,0,0,0)
7	7	(1,0,1,1,1,1,0)

Final partition is achieved and the number of iterations required is 4.

Example 3.3.3. This is an example of a regular tournament in which Step 6 is used (but Step 5 is not used). The tournament T is given by the following adjacency relation.

T : 1(2,3,4); 2(3,4,5); 3(4,5,6); 4(5,6,7); 5(1,6,7);
6(1,2,7); 7(1,2,3)

Vertex label	L(I)	=	1	2	3	4	5	6	7
Score sequence	S(I)	=	3	3	3	3	3	3	3

Since T is regular, no partition of vertex set is possible on the basis of score sequence and therefore we use Step 6.

Iteration No. = 1

Cell No.	Vertices	Lists
1	1	(0,3)
2	2	(0,3)
	3	(0,3)
	4	(0,3)
	5	(1,2)
	6	(1,2)
	7	(1,2)

Iteration No. = 2

Cell No.	Vertices	Lists
1	1	(0,3,0)
2	4 3 2	(0,0,3) (0,1,2) (0,2,1)
3	5 6 7	(1,0,2) (1,1,1) (1,2,0)

Iteration No. = 3

Cell No.	Vertices	Lists
1	1	(0,1,1,1,0,0,0)
2	4	(0,0,0,0,1,1,1)
3	3	(0,1,0,0,1,1,0)
4	2	(0,1,1,0,1,0,0)
5	5	(1,0,0,0,0,1,1)
6	6	(1,0,0,1,0,0,1)
7	7	(1,0,1,1,0,0,0)

Final partition is achieved and the number of iterations required is 3.

Example 3.3.4. This is an example of a regular tournament in which Step 5 is also used to get the final partition. The tournament T is given by the following adjacency relation

T : 1(2,3,5,9); 2(3,4,6,9); 3(4,5,7,9); 4(1,5,6,8); 5(2,6,7,9);
6(1,3,7,8); 7(1,2,4,8); 8(1,2,3,5); 9(4,6,7,8)

Vertex label	L(I) =	1	2	3	4	5	6	7	8	9
Score sequence	S(I) =	4	4	4	4	4	4	4	4	4

No further refinement on the basis of lists associated with vertices of cells is possible and hence we apply Step 5.

Iteration No. = 4

Cell No.	Vertices	Lists
1	3	(0,0,1,1,2,0)
2	1	(1,0,1,2,0,0)
3	9	(0,0,0,0,3,1)
4	5 2	(0,0,1,1,2,0) (1,0,1,0,2,0)
5	7 4 6	(0,1,0,1,1,1) (0,1,0,1,1,1) (1,1,0,0,1,1)
6	8	(1,1,0,2,0,0)

Iteration No. = 5

Cell No.	Vertices	Lists
1	3	(0,0,1,1,0,2,0,0)
2	1	(1,0,1,1,1,0,0,0)
3	9	(0,0,0,0,0,2,1,1)
4	5	(0,0,1,0,1,1,1,0)
5	2	(1,0,1,0,0,1,1,0)
6	7 4	(0,1,0,0,1,1,0,1) (0,1,0,1,0,0,1,1)
7	6	(1,1,0,0,0,1,0,1)
8	8	(1,1,0,1,1,0,0,0)

Iteration No. = 6

Cell No.	Vertices	Lists
1	3	(0,0,1,1,0,1,1,0,0)
2	1	(1,0,1,1,1,0,0,0,0)
3	9	(0,0,0,0,0,1,1,1,1)
4	5	(0,0,1,0,1,1,0,1,0)
5	2	(1,0,1,0,0,0,1,1,0)
6	7	(0,1,0,0,1,0,1,0,1)
7	4	(0,1,0,1,0,0,0,1,1)
8	6	(1,1,0,0,0,1,0,0,1)
9	8	(1,1,0,1,1,0,0,0,0)

Final partition is achieved and the number of iterations required is 6.

3.4 APPLICATION OF THE REFINEMENT PROCEDURE TO TOURNAMENT ISOMORPHISM

Let $T_1 = (V_1, E_1)$ and $T_2 = (V_2, E_2)$ be two tournaments to be tested for isomorphism. Let S_1 and S_2 be the score sequences of tournaments T_1 and T_2 respectively. A necessary condition for $T_1 \cong T_2$ is $S_1 = S_2$. If $S_1 = S_2$, then T_1 and T_2 have the same number of strong components. Let $T_i = [T_{i1}, T_{i2}, \ldots, T_{ik}]$ be the unique strong component decomposition of the tournament T_i into its strong components for $i = 1,2$. Let S_{ij} be the strong score sequence of the strong component T_{ij} for $i = 1,2$ and $j = 1,2,\ldots,k$. We know that $T_1 \cong T_2$ iff $T_{1j} \cong T_{2j}$ for $j = 1,2,\ldots,k$. To each T_{1j} and T_{2j} the following technique is applied if S_{1j} is not simple.

When S_{1j} is simple, $T_{1j} \cong T_{2j}$ and in this case the refinement procedure need not be applied.

We apply refinement procedure on T_{1j} and T_{2j}. After arranging the vectors in antilexicographic order in Step 3 of 3.3.1, we check whether the lists (vectors) associated with the vertices of each cell are identical for T_{1j} and T_{2j} or not. If Step 5 or 6 is not used, then $T_{1j} \not\cong T_{2j}$ and hence $T_1 \not\cong T_2$. So we stop.

Suppose at any stage of refinement procedure 3.3.1, Step 5 is used. Let the first cell be the ith cell in T_{1j} which has got more than one vertices and has not been further refined. We choose the first vertex of ith cell in T_{1j} and place it in the first cell (i.e. Step 5 of 3.3.1 is applied). We choose the first vertex of the ith cell of T_{2j} and place it in the first cell (again Step 5 of 3.3.1 is applied). If the lists associated with vertices of each cell of T_{1j} after using Step 3 of 3.3.1 are identical, then we proceed with the refinement procedure. If not, then we place the vertex of the first cell of T_{2j} in the ith cell in its original position and the next vertex of the ith cell of T_{2j} in the first cell. We now apply the refinement procedure. If still the lists associated with the vertices of each cell of T_{1j} and T_{2j} are not identical after applying Step 3 of refinement procedure, we place the vertex of Ist cell of T_{2j} in the ith cell in its original position and the next vertex of the ith cell of T_{2j} in the first cell. We now apply the refinement procedure. We

continue till all the vertices of the ith cell of T_{2j} have been placed one by one in the first cell. If at no stage the lists associated with vertices of each cell of T_{1j} and T_{2j} are identical, then $T_{1j} \ncong T_{2j}$ and hence $T_1 \ncong T_2$.

If Step 6 is used at any stage, then the same procedure is adopted as for Step 5. The refinement procedure shows that $T_{1j} \cong T_{2j}$ or not.

3.5 TOURNAMENT ISOMORPHISM ALGORITHM

Let $T_1 = (V_1, E_1)$ and $T_2 = (V_2, E_2)$ be two tournaments to be tested for isomorphism. The tournament isomorphism technique is as follows.

Step 1. We calculate the score sequences S_1 and S_2 of the tournaments T_1 and T_2 respectively. If $S_1 \neq S_2$, then $T_1 \ncong T_2$ and we stop, Otherwise we go to Step 2.

Step 2. We calculate the strong components $T_{i1}, T_{i2}, \ldots, T_{ik}$ (unique decomposition) of the tournament T_i for $i = 1,2$ respectively. We also calculate the strong score sequence S_{ij} of the strong component T_{ij} for $i = 1,2$, and $j = 1,2,\ldots,k$. If each S_{1j} is simple then S_1 is simple implying $T_1 \cong T_2$ and we stop. Otherwise we go to Step 3.

Step 3. We follow the refinement procedure of tournament for each pair of strong components T_{1j} and T_{2j} (provided S_{1j} is not simple (As S_{1j} simple implies $T_{1j} \cong T_{2j}$). If $T_{1j} \ncong T_{2j}$ for some j, then $T_1 \ncong T_2$ and we stop. If $T_{1j} \cong T_{2j}$ for $j = 1,2,\ldots,k$, then $T_1 \cong T_2$ and we stop.

We have run the program for the tournament isomorphism on computer upto 20 vertices with almost all possible score sequences and different possible structures of tournaments. Our program worked successfully. This technique helps us decide whether a given pair of tournaments is isomorphic or not.

CHAPTER 4

BIPARTITE TOURNAMENTS

In Chapter 2 we had studied some of the properties of the ordinary tournaments and in Chapter 3 we had discussed the tournament isomorphism problem. Much work has been done on tournaments e.g. [53,54,82,88,107]. It is interesting to see whether a portion of tournament theory can be applied to other areas. In this chapter we shall study one more class of tournaments known as bipartite tournaments. Beineke [9] has made comparisons of the results for the two classes of tournaments, namely ordinary tournaments and bipartite tournaments. This paper also includes a brief bibliography on bipartite tournaments.

The bipartite tournaments can be considered as the result of competition between the individuals of two teams or comparison between the items in two sets. In the case of competition between the individuals of two teams, each player of a team plays with every player of the other team and either this player wins or loses, no tie being allowed. In the event of comparisons between the items in two sets one is asked to give his choice of each item in one set to the every item in the other set. Some more interpretations of the bipartite tournaments can also be given.

Formally a bipartite tournament **T is a complete oriented** bipartite graph. Thus the vertex set is partitioned in to two

disjoint sets $X = \{u_1, u_2, \ldots, u_m\}$ and $Y = \{v_1, v_2, \ldots, v_n\}$. The sets X and Y are known as the partite sets. A bipartite tournament with $|X| = m$ and $|Y| = n$ is known as an mxn-bipartite tournament. The score lists of an mxn-bipartite tournament are the pair of score lists, arranged in nondecreasing order, one for each partite set. The pair of bipartite score lists of an mxn-bipartite tournament will be denoted by $A = (a_1, a_2, \ldots, a_m)$ and $B = (b_1, b_2, \ldots, b_n)$, each list arranged in nondecreasing order.

In Section 4.1, we shall study the bipartite score lists and the strong components of bipartite tournaments as well as the strong components of bipartite score lists. In this section, we also discuss about the total number of bipartite score lists of order mxn and total number of strong bipartite score lists of order mxn. In Section 4.2, we discuss the bipartite tournament isomorphism problem. In Section 4.3, we study the simple pair of bipartite score lists. First we count the number of strong simple pairs of bipartite score lists of order mxn and then we present a recurrence relation to evaluate the value of the total number of simple pairs of bipartite score lists of order mxn. In the last Section 4.4, we partially characterise the self-converse bipartite score lists.

4.1 BIPARTITE SCORE LISTS AND THEIR STRONG COMPONENTS

In this section, we first study what collections of nonnegative integers constitute the score lists of some bipartite tournament T. Then we discuss about the total number of bipartite

score lists of order mxn. In the last, some values of strong bipartite score lists are reported.

Landau [70] has characterised the score sequences of ordinary tournaments (see Theorem 2.1.2). The analogous result in the case of bipartite tournaments has been given by Beineke and Moon [10]. First we report a result which helps in getting a bipartite tournament from the given pair of bipartite score lists.

<u>Theorem 4.1.1</u> [10] . Let $A = (a_1, a_2, \ldots, a_m)$ and $B = (b_1, b_2, \ldots, b_n)$ be two sets of nonnegative integers in nondecreasing order. Let A_1 be obtained from A by deleting one entry a_i and B_1 be obtained from B by reducing $n-a_i$ largest entries of B by 1. Then A and B are the bipartite score lists iff A_1 and B_1 are.

If in Theorem 4.1.1 the last entry of A is deleted and the nondecreasing property of the list B is preserved, then we have a technique for construction of a canonical bipartite tournament with given pair of bipartite score lists. The canonical tournament obtained in this way from bipartite score lists Aand B is denoted by $T^*(A,B)$.

Now we state the existence criteria of bipartite score lists which are analogous to Landau's result [70] in the case of ordinary tournaments for the existence of score sequences. We give a different proof for this result which is much shorter.

<u>Theorem 4.1.2</u> [10] . Let $A = (a_1, a_2, \ldots, a_m)$ and $B = (b_1, b_2, \ldots, b_n)$ be the lists of nonnegative integers in nondecreasing order. Then A and B are the score lists of some bipartite tournament T iff

$$\sum_{i=1}^{k} a_i + \sum_{j=1}^{l} b_j \geq kl \text{ for } 1 \leq k \leq m \text{ and } 1 \leq l \leq n \quad (4.1.1)$$

and

$$\sum_{i=1}^{m} a_i + \sum_{j=1}^{n} b_j = mn \quad (4.1.2)$$

<u>Proof</u>. Let $A = (a_1, a_2, \ldots, a_m)$ and $B = (b_1, b_2, \ldots, b_n)$ be a pair of score lists of a bipartite tournament T of order mxn. Let $X = \{u_1, u_2, \ldots, u_m\}$ and $Y = \{v_1, v_2, \ldots, v_n\}$ be the partite sets of T such that $a_i = s(u_i)$ for $1 \leq i \leq m$ and $b_j = s(v_j)$ for $1 \leq j \leq n$. Consider the subbipartite tournament induced by the vertices $\{u_1, u_2, \ldots, u_k\}$ for $k \leq m$ and $\{v_1, v_2, \ldots, v_l\}$ for $l \leq n$. The number of arcs in this subbipartite tournament is kl. The expression $\sum_{i=1}^{k} a_i + \sum_{j=1}^{l} b_j$ is the sum of the scores of the vertices of this subbipartite tournament and hence

$$\sum_{i=1}^{k} a_i + \sum_{j=1}^{l} b_j \geq kl \text{ for } 1 \leq k \leq m$$

and $1 \leq l \leq n$. For $k = m$ and $l = n$ the sum $\sum_{i=1}^{m} a_i + \sum_{j=1}^{n} b_j$ equals the number of arcs in bipartite tournament T, which is mn. This proves one part.

We prove the other part by contradiction. We assume that $A = (a_1, a_2, \ldots, a_m)$ and $B = (b_1, b_2, \ldots, b_n)$ is a pair of lists

which satisfy equations (4.1.1) and (4.1.2) but are not bipartite score lists. We choose this pair of lists in such a way that m and n are the smallest possible and a_1 is the smallest possible with that choice of m and n. We first consider the case where

$$\sum_{i=1}^{k} a_i + \sum_{j=1}^{l} b_j = kl \text{ for some } k \leq m \text{ and } l < n.$$

By the minimality of m and n, $A_1 = (a_1, a_2, \ldots, a_k)$ and $B_1 = (b_1, b_2, \ldots, b_l)$ is a pair of score lists of some bipartite tournament T_1. Let $A_2 = (a_{k+1}-l, \ldots, a_m-l)$ and $B_2 = (b_{l+1}-k, \ldots, b_n-k)$. We note that

$$\sum_{i=1}^{p} (a_{k+i}-l) + \sum_{j=1}^{q} (b_{l+j}-k)$$

$$= \sum_{i=1}^{k+p} a_i + \sum_{j=1}^{l+q} b_j - \sum_{i=1}^{k} a_i - \sum_{j=1}^{l} b_j - pl - qk$$

$$\geq (k+p)(l+q) - kl - pl - qk$$

$= pq$ for each p, $1 \leq p \leq m-k$ and each q, $1 \leq q \leq n-l$, with the equality holding for $p = m-k$ and $q = n-l$. So by the minimality of m and n, the pair of lists A_2 and B_2 forms the score lists of some bipartite tournament T_2. Let X_i and Y_i be the partite sets of T_i for $1 \leq i \leq 2$. Let $X_1 = \{u_1, u_2, \ldots, u_k\}$, $X_2 = \{u_{k+1}, \ldots, u_m\}$, $Y_1 = \{v_1, v_2, \ldots, v_l\}$ and $Y_2 = \{v_{l+1}, \ldots, v_n\}$. Let $X = X_1 \cup X_2$ and $Y = Y_1 \cup Y_2$. We define a bipartite tournament T with partite sets X and Y. The arcs of T are the arcs of T_1 and T_2 and each u_i dominates every v_j for $k+1 \leq i \leq m$ and $i \leq j \leq l$ and each v_j dominates every u_i for $l+1 \leq j \leq n$ and

$1 \leq i \leq k$. Thus we get a bipartite tournament T with the score lists A and B which is a contradiction.

Now we consider the case when the strict inequality holds in equation (4.1.1) for $k \neq m$ and $l \neq n$. In particular we assume that $a_1 > 0$. Let $A_1 = (a_1-1, a_2, \ldots, a_m+1)$ and $B_1 = (b_1, b_2, \ldots, b_n)$. Clearly the lists A_1 and B_1 satisfy the equations (4.1.1) and (4.1.2). Thus by the minimality of a_1, the lists A_1 and B_1 are a pair of score lists of some bipartite tournament T_1. Let $s(u_1) = a_1-1$ and $s(u_m) = a_m+1$. As $s(u_m) > s(u_1)$ there exists atleast one vertex v such that (u_m, v) and (v, u_1) are the arcs of T_1. By reversing the orientations of the path u_m, v, u_1, we get a bipartite tournament with score lists A and B.||

Another Landau type criteria for the existence of bipartite score lists can be found in Ryser [92] and Beineke and Moon [10].

Let $A = (a_1, a_2, \ldots, a_m)$ and $B = (b_1, b_2, \ldots, b_n)$ be a pair of bipartite score lists of order mxn. Let $A' = (n-a_m, \ldots, n-a_2, n-a_1)$ and $B' = (m-b_n, \ldots, m-b_2, m-b_1)$. Clearly the pair of lists A and B is a pair of bipartite score lists iff the pair A′ and B′ is. The pair of bipartite score lists A′ and B′ is known as the duals of A and B. If A and B are the score lists of a bipartite tournament T then A′ and B′ are the score lists of T′ and the converse also holds good.

In Chapter 3 we have observed that an ordinary tournament can uniquely be decomposed in to its strong components. A

similar observation has been made by Beineke and Moon [10] in the case of bipartite tournaments.

Theorem 4.1.3 [10]. Let $A = (a_1, a_2, \ldots, a_m)$ and $B = (b_1, b_2, \ldots \ldots, b_n)$ be the bipartite score lists in nonincreasing order of some bipartite tournament T. Then

(I) If $a_1 = n$ or $b_1 = m$, then the corresponding vertex constitutes a trivial component.

(II) Otherwise, if $k \geq 2$ is the minimum index for which

$$\sum_{i=1}^{k} a_i = \sum_{j=1}^{n} \min(k, m-b_j) \tag{4.1.3}$$

Then a component consists of those vertices in X with scores $a_1, a_2, \ldots, a_k$ and those vertices in Y with scores $> m-k$.

We illustrate this with the help of the following example.

Example 4.1.1. Consider $A = (4,2,2)$ and $B = (2,1,1,0)$; thus $m = 3$ and $n = 4$. As $a_1 = 4$, this means that the partite set X has a trivial dominating component of score 4. Next we get the lists $(2,2)$ and $(2,1,1,0)$. It follows that Y has a vertex of score 2 forming a trivial component. When this vertex has been deleted we have the score lists $(2,2)$ and $(1,1,0)$. For $k = 1$, we obtain the inequality $2 < 3$, but for $k = 2$, $4 = 4$. Therefore we get a strong component having two vertices from X and two vertices from Y. Finally there is a trivial component of the vertex of Y. The strong component decomposition is shown

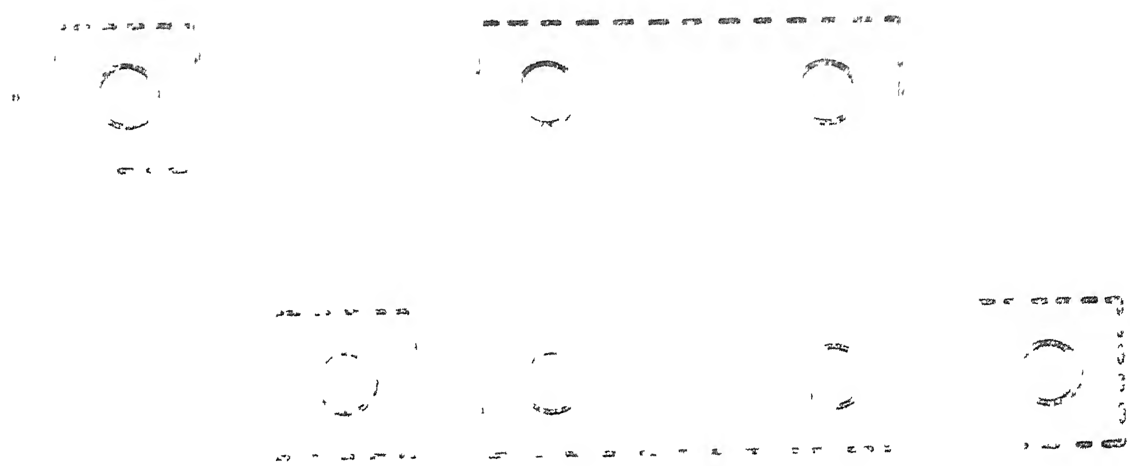
in Figure 4.1.1, where each vertex dominates those vertices in components to the right in the other partite set.

Now we give some definitions.

<u>Definition 4.1.1.</u> A pair of bipartite score lists $A = (a_1, a_2, \ldots, a_m)$ and $B = (b_1, b_2, \ldots, b_n)$ is said to be realisable by a bipartite tournament T with partite sets $X = \{u_1, u_2, \ldots, u_m\}$ and $Y = \{v_1, v_2, \ldots, v_n\}$ if $a_i = s(u_i)$ for $1 \leq i \leq m$ and $b_j = s(v_j)$ for $1 \leq j \leq n$. Such a T is called the realisation of the pair of bipartite score lists A and B.

<u>Definition 4.1.2.</u> Let $T_1, T_2, \ldots, T_p$ be the bipartite tournaments with disjoint partite sets. Let X_i and Y_i be the partite sets of the bipartite tournament T_i for $1 \leq i \leq p$. Clearly $X_i \cap X_j = \varphi$ and $Y_i \cap Y_j = \varphi$ for $i \neq j$. Let $X = \bigcup_{i=1}^{p} X_i$ and $Y = \bigcup_{j=1}^{p} Y_j$. Now $T = [T_1, T_2, \ldots, T_p]$ denote the bipartite tournament obtained from the bipartite tournaments T_i $(1 \leq i \leq p)$, with partite sets X and Y such that the arcs of T are the arcs of T_i and each vertex of Y_j dominates every vertex of X_i for $j < i$ and each vertex of X_i dominates every vertex of Y_j for $i < j$.

Let $T_1, T_2, \ldots, T_p$ be the strong components of a bipartite tournament T obtained with the help of Theorem 4.1.3. We observe



that the strong components $T_1, T_2, \ldots, T_p$ can be arranged in an ordered sequence $T_1, T_2, \ldots, T_p$ such that $T = [T_1, T_2, \ldots, T_p]$. This is known as the strong component decomposition.

<u>Definition 4.1.3.</u> Let $A = (a_1, a_2, \ldots, a_m)$ and $B = (b_1, b_2, \ldots, b_n)$ be a pair of mxn bipartite score lists and $C = (c_1, c_2, \ldots, c_p)$ and $D = (d_1, d_2, \ldots, d_q)$ be another pair of pxq bipartite score lists. We define

$$A+C = (a_1, a_2, \ldots, a_m, \; n+c_1, \; n+c_2, \ldots, n+c_p)$$

$$B+D = (b_1, b_2, \ldots, b_n, \; m+d_1, \; m+d_2, \ldots, m+c_q).$$

Let T_1 and T_2 be the realisations of the pairs of bipartite score lists C and D and A and B respectively. Then $T = [T_1, T_2]$ is a realisation of the pair of lists A+C and B+D. Thus the pair A+C and B+D is a pair of bipartite score lists. We can show that this operation is associative.

Let A and B be a pair of bipartite score lists and A_i and B_i for $1 \leq i \leq p$ be the strong components of A and B. Then we can arrange the strong components A_i and B_i in such a way that

$$A = A_1 + A_2 + \ldots + A_p$$

and

$$B = B_1 + B_2 + \ldots + B_p.$$

Such a decomposition of A and B into its strong components is known as the strong component decomposition.

In the case of ordinary tournaments, the total number of score sequences $t(n)$ of order n has been evaluated by Narayna

and Bent [85]. Let $t(m,n)$ denote the total number of pairs of bipartite score lists of order $m \times n$. Clearly $t(m,n) = t(n,m)$. There is no way to know the number $t(m,n)$ for different values of m and n. Here we present an exhaustive search technique to generate all the $t(m,n)$ pairs of bipartite score lists of order $m \times n$ and thus we can count, for different values of m and n, the value of $t(m,n)$. If $A = (a_1, a_2, \ldots, a_m)$ and $B = (b_1, b_2, \ldots, b_n)$ are the bipartite score lists arranged in nondecreasing order, then $0 \leq a_i \leq n$ for $1 \leq i \leq m$ and $0 \leq b_j \leq m$ for $1 \leq j \leq n$. There are ${}^{m+n}C_m$ $(= {}^{m+n}C_n)$ lists A and B such that $0 \leq a_1 \leq a_2 \leq \ldots$ $\ldots \leq a_m \leq n$ and $0 \leq b_1 \leq b_2 \leq \ldots, \leq b_n \leq m$. We arrange all these lists A and B in antilexicographic order. Then we pick up the first list of A and apply the criteria of Theorem 4.1.2 on every list of B. If the criteria of Theorem 4.1.2 are satisfied, we note down the corresponding A and B. We apply this technique by considering the list of A one by one. In this way we are able to generate all the $t(m,n)$ bipartite score lists of order $m \times n$. We illustrate this procedure with the help of an example.

<u>Example 4.1.2</u>. Let $m = 2$ and $n = 3$, Then the lists A and B arranged in antilexicographic order are as follows.

Sl.No.	List A	List B
1	(0,0)	(0,0,0)
2	(0,1)	(0,0,1)
3	(0,2)	(0,0,2)
4	(0,3)	(0,1,1)

Sl.No.	List A	List B
5	(1,1)	(0,1,2)
6	(1,2)	(0,2,2)
7	(1,3)	(1,1,1)
8	(2,2)	(1,1,2)
9	(2,3)	(1,2,2)
10	(3,3)	(2,2,2)

The pairs of bipartite score lists of order 2x3 are as follows

Sl.No.	List A	List B
1	(0,0)	(2,2,2)
2	(0,1)	(1,2,2)
3	(0,2)	(1,1,2)
4	(0,3)	(1,1,1)
5	(1,1)	(0,2,2)
6	(1,1)	(1,1,2)
7	(1,2)	(0,1,2)
8	(1,2)	(1,1,1)
9	(1,3)	(0,1,1)
10	(2,2)	(0,0,2)
11	(2,2)	(0,1,1)
12	(2,3)	(0,0,1)
13	(3,3)	(0,0,0)

Therefore $t(2,3) = 13$.

Below, we report the values of $t(m,n)$ for some values of m and n. We hope that these values help in getting a general expression for $t(m,n)$ for all values of m and n. We have obtained all these values on computer.

(I) $t(0,k) = 1$ for all integers $k \geq 1$.

(II) $t(1,k) = k+1$ for all integers $k \geq 0$.

(III)

k =	2	3	4	5	6	7	8	9	10
$t(2,k)$ =	7	13	22	34	50	70	95	125	161

The value of $t(2,k)$ for general k is unknown.

(IV)

k =	3	4	5	6	7	8	9
$t(3,k)$ =	34	76	152	280	482	787	1230

The general value of $t(3,k)$ is not known.

(V)

k =	4	5	6
$t(4,k)$ =	221	557	1264.

The value of $t(4,k)$ for all k is unknown.

Let $t_o(m,n)$ denote the number of pairs of bipartite score lists of order $m \times n$ having atleast one score zero. We obtain the following interesting result.

<u>Theorem 4.1.4</u>. $t_o(m,n) = t(m-1,n) + t(m,n-1)$.

<u>Proof</u>. Let A and B be a pair of bipartite score lists of order $m \times n$. Let the strong component decomposition of A and B be

$$A = A_1 + A_2 + \dots + A_p$$

and

$$B = B_1 + B_2 + \dots + B_p$$

where A_i and B_i are the strong components of the pair of bipartite score lists A and B for $1 \leq i \leq p$. We are interested in those pair of bipartite score lists in which atleast one score is zero. This is possible when either $A_1 = (0)$ i.e. the trivial component or $B_1 = (0)$. If $A_1 = (0)$, then by deleting the vertex corresponding to this component we get a bipartite tournament of order $(m-1)xn$. Thus there are $t(m-1,n)$ pairs of bipartite score lists of order mxn such that $A_1 = (0)$. Similarly there are $t(m,n-1)$ pairs of bipartite score lists of order mxn such that $B_1 = (0)$. Therefore $t_o(m,n) = t(m-1,n) + t(m,n-1)$.||

The above result shows that there are $t(m-1,n)+t(m,n-1)$ pairs of bipartite score lists of order mxn which have receivers.

Let $ts(m,n)$ denote the total number of strong pairs of bipartite score lists of order mxn. No result is known which gives the values of $ts(m,n)$ for all values of m and n. Below, we report the values of $ts(m,n)$ for some values of m and n. We hope that these values help to get a general expression to evaluate $ts(m,n)$ for all values of m and n. We know that each strong pair of bipartite score lists has atleast two entries in each list. We have obtained all these values on computer.

(I)

k =	2	3	4	5	6	7
ts(2,k) =	1	1	2	2	3	3

In general $ts(2,k) = [(k-1)/2]$.

(II)

k	= 3	4	5	6	7	8
$ts(3,k)$	= 4	8	16	27	42	62.

The value of $ts(3,k)$ for general k is unknown.

(III)

k	= 4	5	6	7
$ts(4,k)$	= 28	70	157	309.

But $ts(4,k)$ for general k is not known.

(IV) $ts(5,5) = 238$. The value of $ts(5,k)$ for general k is unknown.

4.2 BIPARTITE TOURNAMENT ISOMORPHISM

We have discussed the tournament isomorphism problem in the previous chapter. In this section, we study the problem of bipartite tournament isomorphism. First, we give two definitions of bipartite tournament isomorphism.

Let T_1 and T_2 be the two bipartite tournaments with partite sets X_1 and Y_1 and X_2 and Y_2 respectively. Let $V_i = X_i \cup Y_i$ for $i = 1,2$.

<u>Definition 4.2.1.</u> We say that $T_1 \cong T_2$ if there exists a bijection f from V_1 onto V_2 such that $(u,v) \in E(T_1)$ whenever $(f(u),f(v)) \in E(T_2)$ (This is the usual definition of isomorphism)

<u>Definition 4.2.2.</u> We say that $T_1 \cong T_2$ if there exist bijections f_1 from X_1 onto X_2 and f_2 from Y_1 onto Y_2 such that $(u,v) \in E(T_1)$ whenever $(f_1(u), f_2(v)) \in E(T_2)$ if $u \in X_1$ and $v \in Y_1$ and $(u,v) \in E(T_1)$ whenever $(f_2(u),f_1(v)) \in E(T_2)$ if $u \in Y_1$ and $v \in X_1$.

The bipartite tournament isomorphism problem is to find if such bijection or bijections exist.

In the definition 4.2.2 it is assumed that $|X_1| \neq |Y_2|$. For the case when $|X_1| = |Y_2|$, the partiele set X_1 can be mapped onto the partite set Y_2 by an isomorphic mapping. Below we show that both the definitions given above are equivalent.

Definition 4.2.1 ==> Definition 4.2.2. Let f be an isomorphism given by the definition 4.2.1. We define $f_1 : X_1 \to X_2$ such that $f_1 = f_{|X_1}$ and $f_2 : Y_1 \to Y_2$ such that $f_2 = f_{|Y_1}$. We note that f_1 and f_2 are the required bijections of definition 4.2.2. ||

Definition 4.2.2 ==> Definition 4.2.1. Let f_1 and f_2 be the bijections as given by the definition 4.2.2. We define $f: V_1 \to V_2$ (where $V_1 = X_1 \cup X_2$ and $V_2 = Y_1 \cup Y_2$) such that $f_{|X_1} = f_1$ and $f_{|X_2} = f_2$. We note that f is an isomorphic mapping.

For the case of ordinary tournaments we observe that the tournament isomorphism problem and the regular tournament isomorphism problem are polynomially time equivalent. Analogous of this result in the case of bipartite tournaments is as given below.

<u>Theorem 4.2.1</u>. The bipartite tournament isomorphism and the regular bipartite tournament isomorphism are polynomially time equivalent problems.

<u>Proof</u>. Let T be a bipartite tournament of order mxn with partite sets X and Y. We take four copies of T namely T_i with partite sets X_i, Y_i for $1 \leq i \leq 4$.

Now from T we construct a regular bipartite tournament R(T) with partite sets X^* and Y^* where $X^* = X_1 \cup Y_2 \cup X_3 \cup Y_4$ and $Y^* = Y_1 \cup X_2 \cup Y_3 \cup X_4$. Clearly $|X^*| = |Y^*| = 2(m+n)$. Let $X = \{u_1, u_2, \ldots, u_m\}$ and $Y = \{v_1, v_2, \ldots, v_n\}$ and let $X_i = \{u_{i1}, u_{i2}, \ldots, u_{im}\}$ for $1 \leq i \leq 4$ and $Y_j = \{v_{j1}, v_{j2}, \ldots, v_{jn}\}$ for $1 \leq j \leq 4$.

The edge set E(R(T)) of R(T) is obtained as follows.

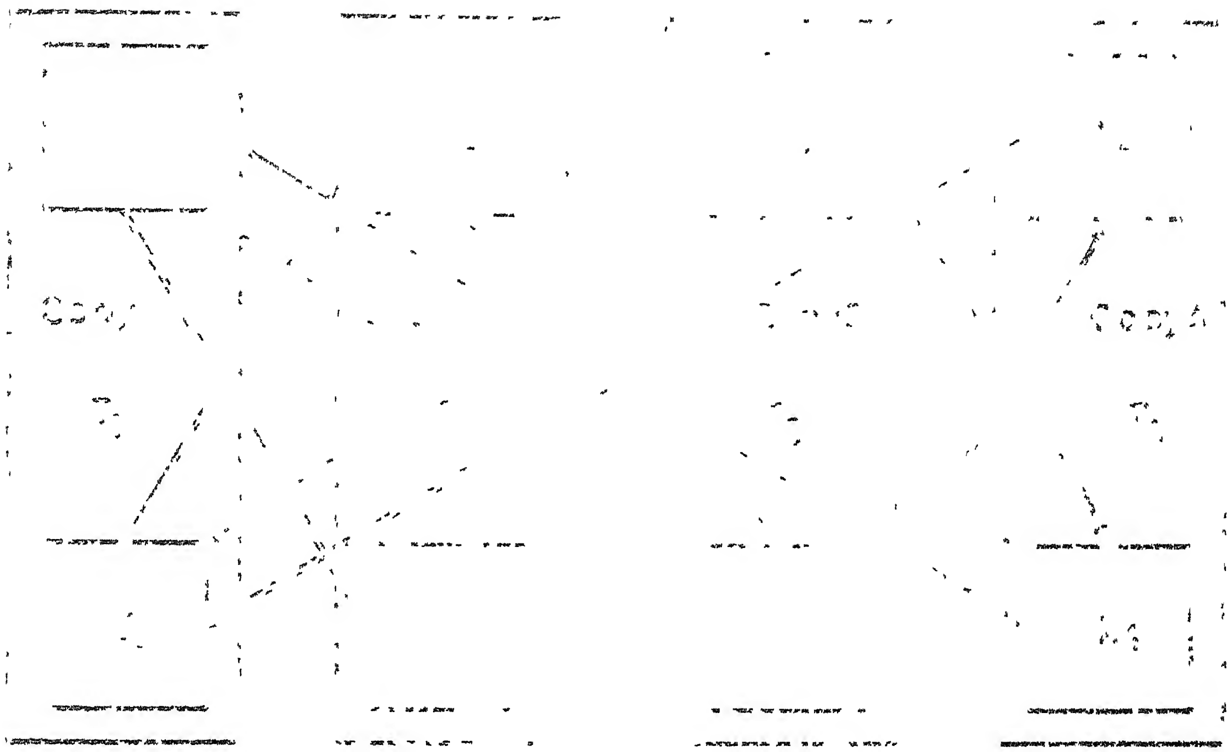
(I) $(u_{ki}, v_{lj}) \in E(R(T))$ for $k \neq l$ if $(v_j, u_i) \in E(T)$ otherwise $(v_{lj}, u_{ki}) \in E(R(T))$.

(II) $(u_{ki}, u_{lj}) \in E(R(T))$ for $1 \leq i, j \leq m$ where $l \equiv k+1 \pmod 4$ i.e. there is a complete orientation from X_k to X_l.

(III) $(v_{ki}, v_{lj}) \in E(R(T))$ for $1 \leq i, j \leq n$. Where $l \equiv k+1 \pmod 4$ i.e. there is a complete orientation from Y_k to Y_l.

A schematic diagram of this construction is shown in Figure 4.2.1. We observe that the bipartite tournament R(T) obtained in this way is regular since $id(v) = od(v) = m+n$ for every vertex v of R(T). Thus we have transformed a bipartite tournament T of order $m \times n$ to a regular bipartite tournament R(T) of order $2(m+n) \times 2(m+n)$.

Let T_1 and T_2 be two bipartite tournaments and $R(T_1)$ and $R(T_2)$ be the regular bipartite tournaments obtained from T_1 and T_2 respectively. It is easy to see that $T_1 \cong T_2$ iff $R(T_1) \cong R(T_2)$. This shows that bipartite tournament isomorphism and regular bipartite tournament isomorphism are polynomially time equivalent problems. ||



$\Longleftrightarrow$: See (i) in the list of edge set

$X \Longrightarrow Y$: Complete , oriented from X to Y

Fig 4.2.1 Schematic diagram -- the construction of regular bipartite tournament .

Now we prove the following result.

Theorem 4.2.2. The bipartite tournament isomorphism problem and the strong bipartite tournament isomorphism problem are polynomially time equivalent.

Proof. Let T_1 and T_2 be the two bipartite tournaments with strong component decompositions as follows.

$$T_1 = [T_{11}, T_{12}, \ldots, T_{1p}] \text{ and } T_2 = [T_{21}, T_{22}, \ldots, T_{2q}]$$

We know that $T_1 \cong T_2$ iff $p = q$ and $T_{1i} \cong T_{2i}$ for $1 \leq i \leq p$. This shows that the bipartite tournament isomorphism problem and strong bipartite tournament isomorphism problem are polynomially time equivalent. ||

We observe that Theorem 4.2.1 becomes a Corollary of Theorem 4.2.2 as every regular bipartite tournament is strong. We note that the proof of Theorem 4.2.1 involves a reduction technique where the proof of Theorem 4.2.2 only uses the structural properties of bipertite tournament.

4.3 SIMPLE PAIRS OF BIPARTITE SCORE LISTS

In Section 2.2 we have studied the simple score sequences of ordinary tournaments. In this section,our aim is to study the simple pairs of bipartite score lists. First we count the number of strong pairs of simple bipartite score lists of order mxn and then we give a recurrence relation to evaluate the number of simple pairs of bipartite score lists of order mxn.

The strong pairs of simple bipartite score lists have been characterised by Beineke and Moon [10].

<u>Theorem 4.3.1</u> (Beineke and Moon [10]). A pair of strong bipartite score lists with atleast three entries in each list is simple iff

(I) One list is $(1,1,\ldots,1)$ or its dual, the other list has no restriction except the strongness, or

(II) One list is $(a,1,\ldots,1)$ with $a > 1$ or its dual and the other list is constant.

The following observation can easily be established and is of much interest.

<u>Lemma 4.3.1</u>. A pair of bipartite score lists is simple iff each of its strong component is simple.

The first observation on simple pair of bipartite score lists is the following.

<u>Theorem 4.3.2</u>. A pair of bipartite score lists A and B is simple iff the dual pair A' and B' is simple.

<u>Proof</u>. Let the pair of bipartite score lists A and B be simple. Suppose that T_1 and T_2 are the two realisations of A' and B' such that $T_1 \not\cong T_2$. This means that $T_1' \not\cong T_2'$. Thus T_1' and T_2' are the two non-isomorphic realisations of a simple pair of bipartite score lists A and B which is a contradiction. Similarly the converse can also be established. ||

The following result gives rise to a new pair of simple bipartite score lists.

<u>Theorem 4.3.3.</u> Let A_i and B_i for $i = 1,2$ be a pair of simple bipartite score lists. Then the pair of bipartite score lists $A = A_1+A_2$ and $B = B_1+B_2$ is also simple.

<u>Proof 1.</u> Let T_i be the realisation of the simple pairs of bipartite score lists A_i and B_i for $i = 1,2$. Then $T = [T_2, T_1]$ is the realisation of A and B. Thus the pair of bipartite score lists A and B is simple. ||

<u>Proof 2.</u> Let the strong component decompositions of the simple pairs of bipartite score lists A_i and B_i for $i = 1,2$ be

$$A_i = A_{i1} + A_{i2} + \dots + A_{ip_i}$$

and

$$B_i = B_{i1} + B_{i2} + \dots + B_{ip_i}$$

Clearly each pair of strong component A_{ij} and B_{ij} for $1 \le i \le 2$ and $1 \le j \le p_i$ is simple. We note that

$$A = A_{11} + A_{12} + \dots + A_{1p_1} + A_{21} + \dots + A_{2p_2}$$

and

$$B = B_{11} + B_{12} + \dots + B_{1p_1} + B_{21} + \dots + B_{2p_2}.$$

Thus the pair of bipartite score lists A and B is simple as each of its strong component is simple. ||

<u>Corollary 4.3.1.</u> Let A_i and B_i for $1 \le i \le q$ be pairs of simple bipartite score lists. Then the pair of bipartite score lists $A = A_1+A_2+\dots+A_q$ and $B = B_1+B_2+\dots+B_q$ is also simple.

<u>Proof.</u> This follows immediately from Theorem 4.3.3. ||

We know that the strong simple score sequences are (0), (1,1,1), (1,1,2,2) and (2,2,2,2,2) (Theorem 2.2.1). Now we shall count the number of strong simple pairs of bipartite score lists of order $m \times n$. First we give some definitions.

Definition 4.3.1. The strong simple pair of bipartite score lists of the kind(I) of Theorem 4.3.1 will be called as the strong simple pair of bipartite score lists of the Ist kind and the strong simple pair of bipartite score lists of the kind(II) of Theorem 4.3.1 as the strong simple pair of bipartite score lists of the IInd kind.

The number of strong simple pairs of bipartite score lists of the Ist and the IInd kinds of order $m \times n$ will be denoted by $H_1(m,n)$ and $H_2(m,n)$ respectively. In our discussion, a strong simple pair of bipartite score lists is either of the Ist kind or of the IInd kind but not of both the kinds. Let $H(m,n)$ denote the number of strong pairs of simple bipartite score lists of order $m \times n$. Clearly

$$H(m,n) = H_1(m,n) + H_2(m,n) \tag{4.3.1}$$

In the subsequent sections, we count the values of $H_1(m,n)$ and $H_2(m,n)$ for all values of m and n. We note that $H(m,n) = H(n,m)$.

4.3.1 NUMBER OF STRONG SIMPLE PAIRS OF BIPARTITE SCORE LISTS OF THE IST KIND

The strong simple pairs of bipartite score lists of the Ist kind are of the form (1,1,...,1) or its dual with no restriction on the other list except the strongness.

Case (I). Let $A = (1,1,\ldots,1)$ or its dual and $B = (b_1,b_2,\ldots,b_n)$ be a strong simple pair of bipartite score lists of the Ist kind of order mxn. Let $H_{11}(m,n)$ denote the number of strong simple pairs of bipartite score lists of this kind of order mxn.

We consider $A = (n-1,n-1,\ldots,n-1)$ and $B = (b_1,b_2,\ldots,b_n)$, a simple pair of strong bipartite score lists of the Ist kind of order mxn. From equation (4.1.2), we get

$$m(n-1) + \sum_{j=1}^{n} b_j = mn$$

i.e.

$$\sum_{j=1}^{n} b_j = m \tag{4.3.2}$$

Thus

$H_{11}(m,n) = 2$(Number of lists $(b_1,b_2,\ldots,b_n)$ satisfying equation (4.3.2) such that the score lists A and B are of the Ist kind) (4.3.3)

The number 2 is appearing in equation (4.3.3) because whenever A and B is a pair of strong simple bipartite score lists of the Ist kind then the pair A′ and B′ is also strong simple bipartite score lists. We note that the list $B = (b_1,b_2,\ldots,b_n)$ satisfies $1 \leq b_1 \leq b_2 \leq \ldots \leq b_n \leq m-1$.

We observe that $n \leq \sum_{j=1}^{n} b_j = m$. Thus for $n > m$ we do not get any list B satisfying equation (4.3.2) and hence

$$H_{11}(m,n) = 0 \text{ for } n > m \tag{4.3.4}$$

If $n = m$, then there is only one list B, namely $(1,1,\ldots,1)$ satisfying equation (4.3.2). In this case $A = (m-1,m-1,\ldots,m-1)$ and $B = (1,1,\ldots,1)$. We note that the duals A' and B' are identical with A and B as $A' = B$ and $B' = A$. Thus

$$H_{11}(m,m) = 1 \tag{4.3.5}$$

Now we consider the case when $n < m$. First, we assign $b_j = 1$ for $1 \leq j \leq n$ and the difference $m-n$ is partitioned into at the most n parts such that the sum is $m-n$ and the pair of lists A and B is of the Ist kind. Thus we get

$$H_{11}(m,n) = 2(p_n(m-n)-1) \text{ for } n < m \tag{4.3.6}$$

where $p_n(m-n)$ is the number of partitions of $m-n$ in to parts not exceeding n. The value of $p_m(k)$ can be obtained from [3, p. 309]. We note this

$$\prod_{k=1}^{m} \frac{1}{1-x^k} = 1 + \sum_{k=1}^{\infty} p_m(k)\ x^k \tag{4.3.7}$$

where $0 \leq x < 1$.

Let $p(k)$ denote the number of ways k can be written as a sum of positive integers $\leq k$. Clearly $p_m(k) = p(k)$ for $m \geq k$. The value of $p(k)$ for $k \geq 0$ can be obtained with the help of Euler's recursion formula [3, Theorem 14.2 or Theorem 14.4]. The result is as follows.

Let $p(0) = 1$ and $p(k) = 0$ if $k < 0$. Then for $k \geq 1$ we have

$$p(k)-p(k-1)-p(k-2)+p(k-5)+p(k-7)+\ldots = 0 \tag{4.3.8}$$

Case (II). Let $A = (a_1, a_2, \ldots, a_m)$ and $B = (1, 1, \ldots, 1)$ or its dual, be a strong simple pair of bipartite score lists of the Ist kind of order mxn. Let $H_{12}(m,n)$ denote the number of strong simple pairs of bipartite score lists of this kind of order mxn. Now a treatment similar to Case (I) yields.

$$H_{12}(m,n) = H_{11}(n,m) \tag{4.3.9}$$

We note that

$$H_1(m,n) = H_{11}(m,n) + H_{12}(m,n) \tag{4.3.10}$$

Therefore

$$H_1(m,n) = H_{11}(m,n) + H_{11}(n,m) \tag{4.3.11}$$

Thus we obtain the following results.

For $n > m$, $H_1(m,n) = 2(p_m(n-m)-1)$ from equations (4.3.4), (4.3.6) and (4.3.11).

For $n = m$, $H_1(m,m) = 2$ from equation (4.3.5).

Finally for $n < m$, $H_1(m,n) = 2(p_n(m-n)-1)$ from equations (4.3.4), (4.3.6) and (4.3.11).

Table 4.3.1 lists $H_1(m,n)$ for some values of m and n for m > n.

k	=	1	2	3	4	5	6	7	8	9	10	11
$H_1(m,m-k)$	=	0	2	4	8	12	20	28	42	58	82	110

Table 4.3.1

4.3.2 NUMBER OF STRONG SIMPLE PAIRS OF BIPARTITE SCORE LISTS OF THE IIND KIND

The strong simple pair of bipartite score lists of the IInd kind is of the form $(a, 1, \ldots, 1)$, $a > 1$ or its dual and

the other list is constant. (Definition 4.3.2).

Case (I). Let $A = (a,1,\ldots,1)$ $a > 1$ or its dual and $B = (t,t,\ldots,t)$ be a strong simple pair of bipartite score lists of the IInd kind of order mxn. Let $H_{21}(m,n)$ denote the number of strong simple pairs of bipartite score lists of order mxn of this kind.

We consider $A = (a,1,\ldots,1)$, $2 \leq a \leq n-1$ and $B = (t,t,\ldots,t)$, $1 \leq t \leq m-1$, a strong simple pair of bipartite score lists of the IInd kind of order mxn. From equation (4.1.2), we get

$$a + m-1 + nt = mn$$

Thus

$$t = m - \frac{m+a-1}{n}, \quad 2 \leq a \leq n-1 \qquad (4.3.12)$$

As a varies from 2 to n-1, we observe that $m+a-1$ gets the $n-2$ consecutive values, namely $m+1$, $m+2$, ..., $m+n-2$. Now we consider a set of n consecutive integers, namely $m-1$, m, $m+1,\ldots,m+n-2$. We know that one and only one of these integers is divisible by n. If $n \nmid m$ and $n \nmid (m-1)$, then from equation (4.3.12) t gets an integer value and the desired pair of bipartite score lists exists. Therefore

$$H_{21}(m,n) = 2 \text{ if } n \nmid m \text{ and } n \nmid (m-1) \qquad (4.3.13)$$

$$= 0 \text{ if } n \mid m \text{ or } n \mid (m-1) \qquad (4.3.14)$$

Case (II). Let $A = (s,s,\ldots,s)$ and $B = (b,1,1,\ldots,1)$, $b > 1$ or its dual, be a strong simple pair of bipartite score lists of the IInd kind of order mxn. Let $H_{22}(m,n)$ denote the number of

strong simple pairs of bipartite score lists of order mxn of this kind. Similar to Case (I), we obtain the following results

$$H_{22}(m,n) = 2 \text{ if } m \nmid n \text{ and } m \nmid (n-1) \tag{4.3.15}$$

$$= 0 \text{ if } m|n \text{ or } m|(n-1) \tag{4.3.16}$$

We note that

$$H_2(m,n) = H_{21}(m,n) + H_{22}(m,n) \tag{4.3.17}$$

Finally we conclude this section with the following results. If $n \nmid m$ and $n \nmid (m-1)$ but $m|n$ or $m|(n-1)$, then $H_2(m,n) = 2$ from equations (4.3.13) and (4.3.16). If $m \nmid n$ and $m \nmid (n-1)$ but $n|m$ or $n|(m-1)$, then $H_2(n,m) = 2$ from equations (4.3.14) and (4.3.15). If $m \nmid n$ and $m \nmid (n-1)$ and $n \nmid m$ and $n \nmid (m-1)$, then $H_2(m,n) = 4$ from equations (4.3.13) and (4.3.15). In the end $H_2(m,m) = 0$ from equations (4.3.13) and (4.3.15).

Thus with the help of Sections 4.3.1 and 4.3.2, we can get the value of $H(m,n)$ for all values of m and n. The values of $H(m,n)$ will be used in the next section for evaluating the total number of simple pairs of bipartite score lists of order mxn.

4.3.3 NUMBER OF SIMPLE PAIRS OF BIPARTITE SCORE LISTS

Let $s(m,n)$ denote the number of simple pairs of bipartite score lists of order mxn. In this section, we evaluate the value of $s(m,n)$ with the help of a recurrence relation. Let $s_o(m,n)$ denote the number of simple pairs of bipartite score lists of order mxn having atleast one score 0. We get the following result.

<u>Theorem 4.3.4.</u> $s_o(m,n) = s(m-1,n) + s(m,n-1)$.

<u>Proof</u>. Let A and B be a simple pair of bipartite score lists of order mxn. Let the strong component decomposition of A and B be

$$A = A_1 + A_2 + \ldots + A_p$$

and

$$B = B_1 + B_2 + \ldots + B_p$$

where the pair A_i and B_i for $1 \leq i \leq p$ is a strong component of A and B and is simple. We are interested in those simple pairs of bipartite score lists for which either $A_1 = (0)$ or $B_1 = (0)$. If $A_1 = (0)$, then the deletion of the vertex corresponding to this score reduces the order of bipartite tournament to $(m-1)$xn. Thus there are $s(m-1,n)$ simple pairs of bipartite score lists of order mxn such that $A_1 = (0)$. Similarly there are $s(m,n-1)$ simple pairs of bipartite score lists of order mxn such that $B_1 = (0)$. Hence

$$s_o(m,n) = s(m-1,n) + s(m,n-1). \quad ||$$

We have observed that all 1xn, mx1, 2xn, mx2 bipartite score lists are simple. We also note that

$$s(m,n) = s(n,m) \tag{4.3.18}$$

The value of $s(m,n)$ can be obtained with the help of the following recurrence relation

$$s(m,n) = \sum_{k=0}^{m} \sum_{l=0}^{n} H(k,l)\ s(m-k,n-l) \tag{4.3.19}$$

where $H(0,0) = 0$; $H(1,0) = 1$; $H(k,0) = 0$ for $k > 1$; $H(k,l) = 0$ for $k > 0$; $s(k,0) = 1$ for $k \geq 0$.

The following tables 4.3.2 and 4.3.3 list the values of $H(k,l)$ for $0 \leq k, l \leq 5$ and $s(m,n)$ for $1 \leq m,n \leq 5$ respectively.

$H(k,l) =$

k \ $l =$	0	1	2	3	4	5
0	0	1	0	0	0	0
1	1	0	0	0	0	0
2	0	0	1	1	2	2
3	0	0	1	2	2	6
4	0	0	2	2	2	2
5	0	0	2	6	2	2

Table 4.3.2

$s(m,n) =$

m \ $n =$	1	2	3	4	5
1	2	3	4	5	6
2	3	7	13	22	34
3	4	13	32	66	124
4	5	22	66	159	339
5	6	34	124	339	804

Table 4.3.3

4.4 SELF-CONVERSE BIPARTITE SCORE LISTS

A pair of $m \times n$ bipartite score lists $A = (a_1, a_2, \ldots, a_m)$ and $B = (b_1, b_2, \ldots, b_n)$ is said to be self-converse if all the bipartite tournaments T with score lists A and B are self-converse

that is $T \cong T'$. In the case of ordinary tournaments the self-converse score sequences have been characterised by Eplett [39] (see Theorem 2.3.1). Eplett's result precisely states that a score sequence S is self-converse iff $S = S'$. Beineke [9] and Beineke and Moon [10] have claimed that a natural analogue of this, in case of bipartite tournaments, must be as follows.

A pair of bipartite score lists $A = (a_1, a_2, \ldots, a_m)$ and $B = (b_1, b_2, \ldots, b_n)$ is self-converse iff

$$a_i + a_{m+1-i} = n \text{ for } 1 \leq i \leq m \tag{4.4.1}$$

and

$$b_j + b_{n+1-j} = m \text{ for } 1 \leq j \leq n \tag{4.4.2}$$

It is equivalent to saying that a pair of bipartite score lists A and B is self-converse iff $A' = A$ and $B' = B$.

Beineke and Moon [10] reported a counter example with lists $A = (1,1,1,3,3,3)$ and $B = (2,2,4,4)$. This satisfies $A' = A$ and $B' = B$ but is not self-converse. Then they claimed that equations (4.4.1) and (4.4.2) are necessary conditions for self-converseness of a pair of bipartite score lists. Below, we report an example which shows that equations (4.4.1) and (4.4.2) are not even necessary conditions.

<u>Example 4.4.1</u>. Consider $A = (1,1,1)$ and $B = (2,2,2)$, a simple pair of bipartite score lists. We note that $A' \neq A$ and $B' \neq B$ i.e. equations (4.4.1) and (4.4.2) are not satisfied. But $A' = B$ and $B' = A$, which shows that the pair of bipartite score lists

is self-converse because if T is the realisation of A and B, then T' is the realisation of A' and B' i.e. of A and B and hence $T \cong T'$.

Thus we note that the equations (4.4.1) and (4.4.2) are the necessary conditions for self-converness when $m \neq n$. We give a result as below.

<u>Theorem 4.4.1</u>. Let $A = (a_1, a_2, \ldots, a_{2p+1})$, $B = (b_1, b_2, \ldots, b_{2q+1})$, $p \neq q$, be a pair of bipartite score lists. Then A and B can not be self-converse.

<u>Proof</u>. Suppose tht the pair of bipartite score lists A and B is self-converse. Since $p \neq q$ and hence equations (4.4.1) and (4.4.2) are satisfied. From equation (4.4.1), we get

$$a_i + a_{2p+2-i} = 2q+1 \quad \text{for } 1 \leq i \leq 2p+1$$

For $i = p+1$, we obtain

$$2a_{p+1} = 2q+1 \quad \text{i.e. } 2 \mid (2q+1),$$

a contradiction. ||

In the above theorem $p \neq q$. For the case $p = q$, the bipartite score lists A and B may be self-converse.

<u>Example 4.4.2</u>. Let $A = (0,2,2)$ and $B = (1,1,3)$. Here we note that $m = n = 3$. We observe that the pair of bipartite score lists A and B is simple with $A' = B$ and $B' = A$. This shows that the pair of bipartite score lists A and B is self-converse.

We have observed in Section 4.2 that a bipartite tournament can uniquely be decomposed in to its strong components. Let T be a bipartite tournament with the unique strong component decomposition, $T = [T_1, T_2, \ldots, T_p]$. Then $T' = [T'_p, \ldots, T'_2, T'_1]$. Thus $T \cong T'$ iff $T'_i \cong T_{p+1-i}$ for $1 \leq i \leq p$.

Let $A = (a_1, a_2, \ldots, a_m)$ and $B = (b_1, b_2, \ldots, b_n)$ be a pair of bipartite score lists. Let the strong component decomposition of A and B be

$$A = A_1 + A_2 + \ldots + A_p \tag{4.4.3}$$

and

$$B = B_1 + B_2 + \ldots + B_p \tag{4.4.4}$$

where A_i and B_i for $1 \leq i \leq p$ are a pair of strong component of A and B. The dual score lists A' and B' will have the following strong component decomposition.

$$A' = A'_p + \ldots + A'_2 + A'_1 \tag{4.4.5}$$

and

$$B' = B'_p + \ldots + B'_2 + B'_1 \tag{4.4.6}$$

We note that

$$A = A' \text{ iff } A'_i = A_{p+1-i} \text{ for } 1 \leq i \leq p \tag{4.4.7}$$

and

$$B = B' \text{ iff } B'_i = B_{p+1-i} \text{ for } 1 \leq i \leq p \tag{4.4.8}$$

For the case, $A' = B$ and $B' = A$, we get

$$A' = B \text{ iff } A'_i = B_{p+1-i} \text{ for } 1 \leq i \leq p \tag{4.4.9}$$

and

$$B' = A \text{ iff } B'_i = A_{p+1-i} \text{ for } 1 \leq i \leq p \tag{4.4.10}$$

score lists A and B having the following adjacency relation

$T : u_1(v_1);\ u_2(v_1);\ u_3(v_2,v_3);\ u_4(v_2,v_4);$
$\quad v_1(u_3,u_4);\ v_2(u_1,u_2);\ v_3(u_1,u_2,u_4);\ v_4(u_1,u_2,u_3).$

We note that $T \not\cong T'$. Therefore the pair of bipartite score lists A and B is not self-converse.

In Theorems 4.4.2 and 4.4.3, one of the lists is constant. Hence it seems that if A or B is constant with $A' = A$ and $B' = B$ (or $A' = B$), then the pair of bipartite score lists A and B is self-converse. But this is not true as the following example shows.

<u>Example 4.4.4.</u> Let $A = B = (5,5,5,5,5,5,5,5,5,5)$. We note that $A' = A$ and $B' = B$. (Also $A' = B$). Let T be a bipartite tournament with the following adjacency relation

$T : u_1(v_1,v_4,v_6,v_9,v_{10});\ u_2(v_3,v_4,v_7,v_8,v_9);\ u_3(v_3,v_5,v_6,v_8,v_{10});$
$\quad u_4(v_2,v_5,v_7,v_9,v_{10});\ u_5(v_1,v_2,v_5,v_7,v_8);\ u_6(v_1,v_2,v_3,v_4,v_6);$
$\quad u_7(v_1,v_3,v_8,v_9,v_{10});\ u_8(v_2,v_3,v_5,v_8,v_{10});\ u_9(v_1,v_2,v_4,v_6,v_7);$
$\quad u_{10}(v_4,v_5,v_6,v_7,v_9);$
$\quad v_1(u_2,u_3,u_4,u_8,u_{10});\ v_2(u_1,u_2,u_3,u_7,u_{10});\ v_3(u_1,u_4,u_5,u_9,u_{10});$
$\quad v_4(u_3,u_4,u_5,u_7,u_8);\ v_5(u_1,u_2,u_6,u_7,u_9);\ v_6(u_2,u_4,u_5,u_7,u_8);$
$\quad v_7(u_1,u_3,u_6,u_7,u_8);\ v_8(u_1,u_4,u_6,u_9,u_{10});\ v_9(u_3,u_5,u_6,u_8,u_9);$
$\quad v_{10}(u_2,u_5,u_6,u_9,u_{10}).$

We have checked that $T \not\cong T'$. Thus the pair of bipartite score lists A and B is not self-converse.

Below, we give techniques which give rise to self-converse bipartite score lists.

Theorem 4.4.4. Let A and B be a simple pair of bipartite score lists of order mxn. Then A+A′ and B+B′ are self-converse bipartite score lists of order 2mx2n with an even number of strong components.

Proof. The pair of bipartite score lists A+A′ and B+B′ is simple from Theorems 4.3.2 and 4.3.3. We note that (A+A′)′ = A+A′ and (B+B′)′ = B+B′. Thus from Theorem 4.4.2, the pair of bipartite score lists A+A′ and B+B′ is self-converse. Clearly the number of components are even. ||

If we do not take the pair of bipartite score lists A and B to be simple, then A+A′ and B+B′ need not be self-converse as the following example shows.

Example 4.4.5. Let A = (1,1,3) and B = (1,2,2,2). We note that A+A′ = (1,1,3,5,7,7) and B+B′ = (1,2,2,2,4,4,4,5). We consider a bipartite tournament T with the score lists A+A′ and B+B′, as given by the following adjacency relation

$$\begin{aligned} T : {} & u_1(v_1);\ u_2(v_1);\ u_3(v_2,v_3,v_4);\ u_4(v_1,v_2,v_3,v_4); \\ & u_5(v_1,v_2,v_3,v_4,v_5,v_6,v_7);\ u_6(v_1,v_2,v_3,v_4,v_6,v_7,v_8); \\ & v_1(u_3),\ v_2(u_1,u_2);\ v_3(u_1,u_2);\ v_4(u_1,u_2);\ v_5(u_1,u_2,u_3,u_4); \\ & v_6(u_1,u_2,u_3,u_4);\ v_7(u_1,u_2,u_3,u_4);\ v_8(u_1,u_2,u_3,u_4,u_5). \end{aligned}$$

We have checked that $T \ncong T'$ and hence A+A′ and B+B′ are not self-converse bipartite score lists.

Theorem 4.4.5. Let A and B be a pair of self-converse bipartite score lists and C and D be a pair of simple bipartite score lists. Then C+A+C′ and D+B+D′ are a pair of self-converse bipartite score lists with atleast three components.

Proof. Let T_1 be the realisation of the simple pair of bipartite score lists C and D. Let T_2 be a self-converse realisation of the self-converse bipartite score lists A and B. Let $T = [T_1', T_2, T_1]$. We observe that T is a realisation of the pair of bipartite score lists C+A+C′ and D+B+D′. We also note that $T' = [T_1', T_2', T_1]$. This shows that $T \cong T'$ as $T_2 \cong T_2'$. Thus T is a self-converse realisation of C+A+C′ and D+B+D′. Since T is an arbitrary choice and hence $T \cong T'$ for all the realisations T of C+A+C′ and D+B+D′. Thus C+A+C′ and D+B+D′ are self-converse. Clearly the pair C+A+C′ and D+B+D′ has atleast three components.||

In the end, we produce an example of bipartite score lists A and B having two realisations T_1 and T_2, where $T_1 \ncong T_2$. Here T_1 is self-converse but T_2 is not self-converse.

Example 4.4.6. Consider A = (1,3,3,5) and B = (1,1,2,2,3,3). Clearly A′ = A and B′ = B. Let T_1 and T_2 be as given below

T_1 : $u_1(v_1)$; $u_2(v_2,v_3,v_4)$; $u_3(v_1,v_2,v_5)$; $u_4(v_1,v_2,v_3,v_4,v_6)$;
$v_1(u_2)$; $v_2(u_1)$; $v_3(u_1,u_3)$; $v_4(u_1,u_3)$; $v_5(u_1,u_2,u_4)$;
$v_6(u_1,u_2,u_3)$.

T_2 : $u_1(v_3)$; $u_2(v_1,v_2,v_6)$; $u_3(v_1,v_2,v_4)$; $u_4(v_1,v_2,v_3,v_4,v_5)$;
$v_1(u_1)$; $v_2(u_1)$; $v_3(u_2,u_3)$; $v_4(u_1,u_2)$; $v_5(u_1,u_2,u_3)$;
$v_6(u_1,u_3,u_4)$.

We have checked that $T_1 \ncong T_2$, $T_1 \cong T_1'$ and $T_2 \ncong T_2'$.

CHAPTER 5

TRIPARTITE TOURNAMENTS

In this chapter our aim is to study one more class of tournaments namely the tripartite tournaments. We shall be doing comparison of the results for the three classes of the tournaments, namely ordinary, bipartite and tripartite tournaments. The research work on tripartite tournaments is in its infancy.

The tripartite tournaments can be considered as a result of competition among the individuals of three teams or comparison among the items in three sets. In the case of competition among the indivisuals of three teams each player of a team plays with every player of the other two teams and this player either wins or loses, no tie being allowed. In the event of comparisons among the items in three sets one is asked to give his preference of each item in one set to the every item in the other two sets.

In an ordinary tournament a nontrivial strong component must have atleast three vertices while in a bipartite tournament a nontrivial strong component must have atleast four vertices, two from each partite set. In a tripartite tournament a non-trivial strong component must have atleast

three vertices, one from each partite set. There is only one ordinary tournament which is consistent (a tournament is consistent if it has no cycle), namely the transitive tournament. In the case of bipartite tournament there are more than one consistent bipartite tournaments of order m x n. Consistent bipartite tournaments have been characterised by Beineke and Moon [10]. Similarly there are more than one consistent tripartite tournaments of order p x q x r. A regular ordinary tournament is of odd order. A regular bipartite tournament is of order 2m x 2m. Whereas a regular tripartite tournament is of order p x p x p. Every non-trivial regular tournament of each kind is strong.

In section 5.1, first we characterise tripartite score lists. After that we study the strong component decomposition of tripartite tournaments. We also present a technique to generate all the tripartite score lists of order p x q x r with the help of computer. In section 5.2 we give a relation between bipartite tournaments and tripartite tournaments. The concept of score vectors is introduced and the score vectors have been characterised, and some of its properties are studied. At the end we study the existence of cycles in the tripartite tournaments and present some conjectures.

5.1 REALISATION OF TRIPARTITE SCORE LISTS.

In this section we shall study what collections of three sets of nonnegative integers constitute the score lists of some tripartite tournaments. Landau [70] has characterised the score sequences of an ordinary tournament (Theorem 2.1.?). Moon [80] and Beineke and Moon [10] have characterised the bipartite score lists (Theorems 4.1.1 and 4.1.2).

Let $A = (a_1, a_2, \ldots, a_p)$; $B = (b_1, b_2, \ldots, b_q)$ and $C = (c_1, c_2, \ldots, c_r)$ be three sets of nonnegative integers in nondecreasing order.

Let $A' = (q+r-a_p, \ldots, q+r-a_2, q+r-a_1)$;

$B' = (p+r-kq, \ldots, p+r-b_2, p+r-b_1)$ and

$C' = (p+q-c_r, \ldots, p+q-c_2, p+q-c_1)$.

The following are equivalent.

A, B and C are the score lists of some tripartite tournament.

A', B' and C' are the score lists of some tripartite tournament.

A', B' and C' are known as the duals of the tripartite score lists A, B and C respectively.

Definition 5.1.1. Let $A = (a_1, a_2, \ldots, a_p)$, $B = (b_1, b_2, \ldots, b_q)$ and $C = (c_1, c_2, \ldots, c_r)$ be a collection of tripartite score lists. The tripartite score lists A, B and C are said to be **realisable** by a tripartite tournament T with partite sets $X = \{u_1, u_2, \ldots, u_p\}$; $Y = \{v_1, v_2, \ldots, v_q\}$ and $Z = \{w_1, w_2, \ldots, w_r\}$

if $a_i = s(u_i)$ for $1 \le i \le p$, $b_j = s(v_j)$ for $1 \le j \le q$ and $c_k = s(w_k)$ for $1 \le k \le r$. Such a T is called the realisation of A,B and C.

A tripartite tournament is a realisation of tripartite score lists A,B and C iff T' is a realisation of A', B' and C'.

Now we state a result which can be used to obtain a construction of tripartite tournament with the given score lists. This result is an analogue of Theorem 2.1.1 in the case of ordinary tournaments and that of 4.1.1 in case of bipartite tournaments.

Theorem 5.1.1. Let $A = (a_1, a_2, \ldots, a_p)$; $B = (b_1, b_2, \ldots, b_q)$ and $C = (c_1, c_2, \ldots, c_r)$ be the three sets of nonnegative integers in nondecreasing order. Let A_1 be obtained from A by deleting one entry a_1 and B_1 and C_1 be obtained from B and C respectively by reducing $q+r-a_1$ largest entries of B and C by 1. Then A,B and C are the score lists of some tripartite tournament iff A_1, B_1 and C_1 are.

Proof. Let A_1, B_1 and C_1 be the score lists of some tripartite tournament T_1 with partite sets X,Y and Z. To the partite set X we add one vertex u. The vertex u dominates those vertices of Y and Z whose scores were not changed during the transformation of A,B and C to A_1, B_1 and C_1 respectively, and is dominated by all other vertices of Y and Z. Thus we get a tournament with the score lists A,B and C.

Conversely let A,B and C be the score lists of some tripartite tournament T with the partite sets X,Y and Z. We shall show that the vertex u_i of X with the score a_i is dominated by the **vertices** of Y and Z with the largest scores. Let this be not the case i.e. Let there exist two vertices x and y such that (u_i,x) and (y,u_i) are the arcs of T and $s(y) < s(x)$. We have the following four possibilities.

(1) $x,y \varepsilon X$, (2) $x \varepsilon Y$ and $y \varepsilon Z$, (3) $x \varepsilon Z$ and $y \varepsilon Y$ and (4) $x,y \varepsilon Z$.

(1) Let $x,y \varepsilon Y$. Since $s(y) < s(x)$ this implies that there exists a vertex z such that (z,y) and (x,z) are the arcs of T. Either $z \varepsilon X$ or $z \varepsilon Z$. Let $z \varepsilon X$, we get a 4-cycle u_i, x,z,y,u_i. By reversing the orientations of the arcs of this 4-cycle we get a tripartite tournament T with the same score lists A,B and C, but the sum of the scores of the vertices dominating u_i is larger than the previous sum. Continuing in this way we get the result. Let $z \varepsilon Z$, again we have a 4-cycle u_i,x,z,y,u_i. The same argument works here.

(2) Let $x \varepsilon Y$ and $y \varepsilon Z$. Since $s(y) < s(x)$ this implies that there exists a vertex z such that (z,y) and (x,z) are the arcs. The vertex z has to be in X ($z \notin Y$ as (x,z) is an arc and $z \notin Z$ as (z,y) is an arc). We get a 4-cycle u_i,x,z,y,u_i. The same argument is applicable here as in the case (1).

(3) Let $x \in Z$ and $y \in Y$. Since $s(y) < s(x)$. This implies that there exists a vertex z such that (z,y) and (x,z) are the arcs of T. The vertex z has to be in $X(z \notin Y$ as (z,y) is an arc ~~and $z \notin Z$ as (x,z) is an arc~~ and $z \notin Z$ as (x,z) is an arc). We get a 4-cycle u_1,x,z,y,u_1. The same reasoning is applied here as in the case (1).

(4) Let $x,y \in Z$. Since $s(y) < s(x)$, this implies that there exists a vertex z such that (z,y) and (x,z) are the arcs of T. Either $z \in X$ or $z \in Y$. Let $z \in X$, we get a 4-cycle u_i,x,z,y,u_1. The same argument works here as in the case (1). If $z \in Y$. Then we have a 4-cycle u_i,x,z,y,u_1. We use the same argument as in the case (1). This completes the proof.||

This theorem is used to obtain a canonical tripartite tournament with the score lists A, B and C.

We begin with our lists A,B and C in nondecreasing order and then delete the last entry of A and maintain the non-decreasing property of B and C while reducing entries untill all the entries of A are deleted. Now we delete the last entry of B and maintain the nondecreasing property of C while reducing the entries of C until all the entries of B are deleted. This procedure results in a canonical tripartite tournament with the score lists A,B and C and is denoted by $T^*(A,B,C)$.

We explain the procedure with the help of the following example.

Example 5.1.1

Let $A = (2,2)$, $B = (4,5)$, $C = (1,1,1)$

$A_1 = (2)$, $B_1 = (3,4)$, $C_1 = (0,1,1)$

$A_2 = \varphi$, $B_2 = (2,3)$, $C_2 = (0,0,1)$

$B_3 = (2)$, $C_3 = (0,0,1)$

$B_4 = \varphi$, $C_4 = (0,0,0)$.

$T^*(A,B,C)$ is given by the following adjacency relation.

$T^*(A,B,C)$: $u_1\ (w_2,w_3)$; $u_2(w_1,w_3)$; $v_1(w_1,w_2,u_1,u_2)$;

$v_2\ (w_1,w_2,w_3,u_1,u_2)$; $w_1(u_1)$; $w_2(u_2)$; $w_3(v_1)$.

If in Theorem 5.1.1 the list C is omitted, then we get a result for bipartite tournaments which have been observed by Beineke and Moon [10] (see Theorem 4.1.1).

It can be easily seen that the canonical tournament $T^*(A,B)$ of the bipartite score lists A and B is unique, while $T^*(A,B,C)$ of the tripartite score lists A,B and C need not be unique. The following example illustrates it.

Example 5.1.2.

Let $A = (1,1)$, $B = (2,3)$, $C = (3,3,3)$

$A_1 = (1)$, $B_1 = (2,2)$, $C_1 = (2,2,2)$

$A_2 = \varphi$, $B_2 = (1,2)$, $C_2 = (1,1,1)$

$B_3 = (1)$, $C_3 = (0,1,1)$

$B_4 = \varphi$, $C_4 = (0,0,0)$.

The canonical tournament $T_1^*(A,B,C)$ is given by the following adjacency relation.

$T_1^*(A,B,C)$: $u_1(v_2)$; $u_2(v_1)$; $v_1(u_1,w_1)$; $v_2(u_2,w_2,w_3)$;
$w_1(u_1,u_2,v_2)$; $w_2(u_1,u_2,v_1)$; $w_3(u_1,u_2,v_1)$.

We consider another reduction as follows

$$\begin{array}{llll} A = (1,1) & , B = (2,3), & C = (3,3,3) \\ A_1 = (1) & , B_1 = (2,2), & C_1 = (2,2,2) \\ A_2 = \varphi & , B_2 = (1,1), & C_2 = (1,1,2) \\ & B_3 = (1) & , C_3 = (0,1,1) \\ & B_4 = \varphi & , C_4 = (0,0,0). \end{array}$$

The canonical tournament $T_2(A,B,C)$ is given by the following adjacency relation.

$T_2^*(A,B,C)$: $u_1(w_3)$; $u_2(v_1)$; $v_1(u_1,w_1)$; $v_2(u_1,u_2,w_3)$;
$w_1(u_1,u_2,v_2)$; $w_2(u_1,u_2,v_1)$; $w_3(u_2,v_1,v_2)$.

We have checked that $T_1^* \not\cong T_2^*$.

Now we prove a result which shows what collection of three sets of nonnegative integers form the score lists of some tripartite tournament. This result is similar to that of Landau's result [70] (Theorem 2.1.2) for ordinary tournaments and Beineke and Moon's [10] (Theorem 4.1.2) for bipartite tournaments.

<u>Theorem 5.1.2</u>. Let $A = (a_1,a_2,\ldots,a_p)$, $B = (b_1,b_2,\ldots,b_q)$ and $C = (c_1,c_2,\ldots,c_r)$ be the three sets of nonnegative

integers in a nondecreasing order. The lists A, B and C form the score lists of some tripartite tournament T iff

(1) $$\sum_{i=1}^{l} a_i + \sum_{j=1}^{m} b_j + \sum_{k=1}^{n} c_k \geq lm + mn + nl \qquad (5.1.1)$$

for $1 \leq l \leq p$, $1 \leq m \leq q$ and $1 \leq n \leq r$, equality holds for $l = p$, $m = q$ and $n = r$

(2) If O is the number of zeros in any lists, then the first entry of the other two lists has to be atleast O.

(3) The condition (2) is satisfied by the duals A′, B′ and C′.

<u>Proof</u>. Let T be a realisation of score lists A, B and C. We show that all the three conditions are satisfied.

(1) Consider the subtripartite tournament with vertices $\{u_1, u_2, \ldots, u_l\}$, $\{v_1, v_2, \ldots, v_m\}$ and $\{w_1, w_2, \ldots, w_n\}$. The number of arcs in this subtripartite tournaments are $lm+mn+nl$. The sum $\sum_{i=1}^{l} a_i + \sum_{j=1}^{m} b_j + \sum_{k=1}^{n} c_k$ is the sum of the score of the vertices $\{u_1, u_2, \ldots, u_l\}$, $\{v_1, v_2, \ldots, v_m\}$ and $\{w_1, w_2, \ldots, w_n\}$. Hence $\sum_{i=1}^{l} a_i + \sum_{j=1}^{m} b_j + \sum_{k=1}^{n} c_k \geq lm + mn + nl$. For $l=p$, $m=q$ and $n=r$ the left hand side of equation (5.1.1) is the sum of the scores of the vertices of T and hence equals the number of arcs in T. This proves the condition (1).

(2) Let $a_1 = a_2 = \ldots = a_O = o$, i.e., the vertices $\{u_1, u_2, \ldots u_O\}$ are dominated by every vertex of Y and Z and hence $b_1 \geq O$ and $c_1 \geq O$.

(3) This can be proved similar to the condition (2).

This proves the necessary part. The sufficient part we prove with the help of contradiction. Let A, B and C be the three sets of non-negative integers in nondecreasing order satisfying conditions (1), (2) and (3) but A, B and C do not form the score lists of any tripartite tournament. We choose these lists such that p, q and r are the smallest possible and a_1 is the smallest possible for the choice of p, q and r.

Case (1). Suppose equality in equation (5.1.1) holds for some $l < p$, $m \le q$ and $n \le r$ i.e.

$$\sum_{i=1}^{l} a_i + \sum_{j=1}^{m} b_j + \sum_{k=1}^{n} c_k = lm+mn+nl \tag{5.1.2}$$

Let $A_1 = (a_1, a_2, \ldots, a_l)$, $B_1 = (b_1, b_2, \ldots, b_m)$ and $C_1 = (c_1, c_2, \ldots, c_n)$. Clearly A_1, B_1 and C_1 satisfy the conditions (1), (2) and (3). Thus by the minimality of p, q and r the lists A_1, B_1 and C_1 are the score lists of some tripartite tournament T_1. Now we define

$A_2 = (a_{l+1}-m-n, \ldots, a_p-m-n)$,

$B_2 = (b_{m+1}-n-l, \ldots, b_q-n-l)$ and

$C_2 = (c_{n+1}-l-m, \ldots, c_r-l-m)$.

We consider the sum

$$\sum_{i=1}^{L} (a_{l+i}-m-n) + \sum_{j=1}^{M} (b_{m+j}-n-l) + \sum_{k=1}^{N} (c_{n+k}-l-m)$$

$$= \sum_{i=1}^{l+L} a_i + \sum_{j=1}^{m+M} b_j + \sum_{k=1}^{n+N} c_k - \left(\sum_{i=1}^{l} a_i + \sum_{j=1}^{m} b_j + \sum_{k=1}^{n} c_k \right)$$

$-Lm-Ln-Ml-Mn-Nl-Nm \geq LM+MN+NL$ for $1 \leq L \leq p-l$, $1 \leq M \leq q-m$ and $1 \leq N \leq r-n$. Equality holds for $L = p-l$, $M = q-m$ and $N = r-n$. Thus the lists A_2, B_2 and C_2 satisfy condition (1) of the theorem.

Now we prove that the condition (2) is satisfied by the lists A_2, B_2 and C_2. Let $a_{l+1} = a_{l+2} = \dots = a_{l+O} = m+n$. Then we have to show that $b_{m+1} - l - n \geq 0$ and $c_{n+1} - l - m \geq 0$. We consider

$$\sum_{i=1}^{l+O} a_i + \sum_{j=1}^{m+1} b_j + \sum_{k=1}^{n} c_k \geq (l+O)\,(m+1)$$

$$+ (m+1)n + n(l+O) \text{ from equation (5.1.1).}$$

Thus

$$\sum_{i=1}^{l} a_i + \sum_{j=1}^{m} b_j + \sum_{k=1}^{n} c_k + O(m+n) + b_{m+1}$$

$$\geq lm + l + Om + O + mn + n + nl + nO$$

i.e. $b_{m+1} \geq l+n+O$. Similarly we can show that $c_{n+1} \geq l+m+O$. Thus the lists A_2, B_2 and C_2 satisfy the condition (2) of theorem. Similarly we can show that A_2', B_2' and C_2' also satisfy the condition (2) of theorem. Hence by the minimality of p, q and r, the lists A_2, B_2 and C_2 form the score lists of some tripartite tournament T_2.

Let X_1, Y_1 and Z_1 be the partite sets of the tripartite tournaments T_1 with the score lists A_i, B_i and C_1 for $1 = 1,2$. Let $X = X_1 \cup X_2$, $Y = Y_1 \cup Y_2$ and $Z = Z_1 \cup Z_2$. We define a tripartite tournament with partite sets X,Y and Z as follows.

Each vertex of X_2 dominates every vertex of Y_1 and Z_1. Each vertex of Y_2 dominates every vertex of X_1 and Z_1. Each vertex of Z_2 dominates every vertex of X_1 and Y_1. Thus we get a tripartite tournament T with the score lists A,B,C which is a contradiction.

Case (2). Suppose that the strict inequality holds in equation (5.1.1) for $1 \neq p$, $m \neq q$ and $n \neq r$. We assume that $a_1 \neq 0$. Let $A_1 = (a_1-1, a_2, \ldots, a_p+1)$, $B_1 = (b_1, b_2, \ldots, b_q)$, $C_1 = (c_1, c_2, \ldots, c_r)$. Clearly the lists A_1, B_1 and C_1 satisfy all the three conditions of the theorem. Hence by the minimality of a_1, the lists A_1, B_1 and C_1 are the score lists of some tripartite tournament T_1. Let $s(u_1) = a_1-1$ and $s(u_p) = a_p+1$. As $s(u_p) > s(u_1)$, there exists a vertex $x \in V(T_1)$ such that (u_p, x) and (x, u_1) are the arcs of T_1. By reversing the orientations of the arcs of the path u_p, x, u_1, we get a tripartite tournament T with the score lists A, B and C, a contradiction.||

Definition 5.1.2. Let T_i be the tripartite tournaments with disjoint partite sets X_i, Y_i and Z_i for $1 \leq i \leq t$.

Let $X = \bigcup_{i=1}^{t} X_i$, $Y = \bigcup_{i=1}^{t} Y_i$ and $Z = \bigcup_{i=1}^{t} Z_i$. Now $T=[T_1, T_2, \ldots, T_t]$ denote the tripartite tournament with the partite sets X,Y and Z, obtained from T_i for $1 \leq i \leq t$ such that the arcs of T are the arcs of T_i and each vertex of X_i dominates every vertex of Y_j and Z_k for $i > j$ and $i > k$, each vertex of Y_j dominates every vertex of X_i and Z_k for $j > i$ and $j > k$ and each vertex of Z_k dominates every vertex of X_i and Y_j for $k > i$ and $k > j$.

<u>Definition 5.1.3</u>. Let $A_i = (a_{i1}, a_{i2}, \ldots, a_{ip_i})$, $B_i = (b_{i1}, b_{i2}, \ldots, b_{iq_i})$ and $C_i = (c_{i1}, c_{i2}, \ldots, c_{ir_i})$ for $i = 1,2$ be some tripartite score lists. We define

$$A_1 + A_2 = (a_{11}, a_{12}, \ldots, a_{1p_1}, q_1+r_1+a_{21}, \ldots, q_1+r_1+a_{2p_2})$$

$$B_1 + B_2 = (b_{11}, b_{12}, \ldots, b_{1q_1}, p_1+r_1+b_{21}, \ldots, p_1+r_1+b_{2q_2})$$

and

$$C_1 + C_2 = (c_{11}, c_{12}, \ldots, c_{1r_1}, p_1+q_1+c_{21}, \ldots, p_1+q_1+c_{2r_2}).$$

Let T_i be a realisation of tripartite score lists A_i, B_i and C_i for $i = 1,2$. Then $T = [T_1, T_2]$ is a realisation of $A_1 + A_2$, $B_1 + B_2$ and $C_1 + C_2$. Thus $A_1 + A_2$, $B_1 + B_2$ and $C_1 + C_2$ are the tripartite score lists. This operation is associative. If A_i, B_i and C_i for $1 \leq i \leq t$ are the tripartite score lists, then $A = A_1 + A_2 + \ldots + A_t$, $B = B_1 + B_2 + \ldots + B_t$ and $C = C_1 + C_2 + \ldots + C_t$ are also the tripartite score lists.

The strong components of tripartite tournament (score lists) can be obtained with the help of the following technique.

(I) Let $A = (a_1, a_2, \ldots, a_p)$, $B = (b_1, b_2, \ldots, b_q)$ and $C = (c_1, c_2, \ldots, c_r)$ be the tripartite score lists. Let T be a realisation of A, B and C. If $a_1 = 0$ or $b_1 = 0$ or $c_1 = 0$, then the corresponding vertex of T constitutes the trivial component. We delete this vertex as well as the entry of the list corresponding to this vertex and scores of other two lists are reduced by 1. Now we consider the new lists and repeat the step. If step (I) is not applicable we go to step (II).

(II) We check for what values of l, m and n the equality holds good in equation (5.1.1), i.e. $\sum_{i=1}^{l} a_i + \sum_{j=1}^{m} b_j + \sum_{k=1}^{n} c_k = lm + mn + nl$ for $1 \leq l \leq p$, $1 \leq m \leq q$ and $1 \leq n \leq r$.

Then the vertices corresponding to the score lists $(a_1, a_2, \ldots, a_l)$, $(b_1, b_2, \ldots, b_m)$ and $(c_1, c_2, \ldots, c_n)$ form a strong component of T. We delete these vertices and the corresponding entries from the score lists. We reduce each entry of A by $m+n$, of B by $l+n$ and of C by $l+m$. We consider this new lists and we go to step (I).

Thus we can get the strong component of a tripartite tournament as well as the tripartite score lists. We consider one example.

Example 5.1.3. Let $A = (1,1)$, $B = (1,5)$ and $C = (2,3,3)$. Equation (5.1.1) is satisfied for $l=2$, $m=1$ and $n=1$. Thus the first strong component is $A_1 = (1,1)$, $B_1 = (1)$ and $C_1 = (2)$. The reduced lists are $A = \varphi$, $B = (2)$ and $C = (0,0)$. Thus there are two trivial components $C_2 = C_3 = (0)$ and one trivial component $B_4 = (0)$.

Let $T_1, T_2, \ldots, T_t$ be the strong components of a tripartite tournament T obtained from the above procedure. Now $T_1, T_2, \ldots, T_t$ can be arranged in an ordered sequence such that $T = [T_1, T_2, \ldots, T_t]$, and such a decomposition is known as the strong component decomposition of T. In the same way the strong components A_i, B_i and C_i for $1 \leq i \leq t$ of the tripartite score lists A, B and C can be arranged in an ordered sequence such that

$$A = A_1 + A_2 + \ldots + A_t$$

$$B = B_1 + B_2 + \ldots + B_t \tag{5.1.3}$$

and

$$C = C_1 + C_2 + \ldots + C_t \ .$$

In section 4.1 we have studied about generating the bipartite score lists with the help of computer and we have also discussed about $t(m,n)$, the total number of bipartite score lists of order $m \times n$. Here we shall be discussing about generating the tripartite score lists of order $p \times q \times r$. We shall study about the number $t(p,q,r)$,

which is the total number of tripartite score lists of order $p \times q \times r$. No exact result is known to evaluate $t(p,q,r)$ for all values of p,q and r. We shall be reporting the values of $t(p,q,r)$ for some values of p,q and r. We hope that these values may help in getting a general expression for $t(p,q,r)$. Now we present the exhaustive search technique. If $A = (a_1,a_2,\ldots,a_p)$, $B = (b_1,b_2,\ldots,b_q)$ and $C = (c_1,c_2,\ldots,c_r)$ are tripartite score lists of order $p \times q \times r$, then $0 \leq a_i \leq q+r$ for $1 \leq i \leq p$, $0 \leq b_j \leq r+p$ for $1 \leq j \leq q$ and $0 \leq c_k \leq p+q$ for $1 \leq k \leq r$. By computer we generate three lists of nonnegative integers in nondecreasing order, $A = (a_1,a_2,\ldots,a_p)$ $0 \leq a_i \leq q+r$ for $1 \leq i \leq p$, $B=(b_1,b_2,\ldots,b_q)$, $0 \leq b_j \leq r+p$ for $1 \leq j \leq q$ and $C = (c_1,c_2,\ldots,c_r)$ $0 \leq c_k \leq p+q$ for $1 \leq k \leq r$ and arrange all these lists in antilexicographic order. Now we choose the first list of A and apply the criteria of the Theorem 5.1.2 by choosing the list of B and C one by one. If the criteria of Theorem 5.1.2 are satisfied for some lists A,B and C, then we note them down. We do this for all the lists of A, choosing them one by one. Thus we get all the $t(p,q,r)$ tripartite score lists of order $p \times q \times r$. In table 5.1.1 we list the values of $t(p,q,r)$ for some values of p,q and r.

p	q	r	t (p,q,r)
1	1	1	7
1	1	2	16
1	1	3	30
1	1	4	50
1	1	5	77
1	1	6	112
1	1	8	210
1	1	9	275
1	2	2	53
1	2	3	136
1	2	4	298
1	2	5	584
2	2	2	239
2	2	3	799

Table 5.1.1

Let $t_o(p,q,r)$ denote the number of tripartite score lists of order $p \times q \times r$ such that atleast one score is zero. We have the following interesting result.

Theorem 5.1.3. $t_o(p,q,r) = t(p-1,q,r) + t(p,q-1,r) + t(p,q,r-1)$

Proof. Let A,B and C be the tripartite score lists of order $p \times q \times r$. Let the strong component decomposition of A, B and C be as follows.

$A = A_1 + A_2 + \ldots + A_t$

$B = B_1 + B_2 + \ldots + B_t$

and

$C = C_1 + C_2 + \ldots + C_t$

where A_i, B_i and C_i for $1 \leq i \leq t$ are the strong components of A, B and C. We are interested only in those tripartite score lists in which either $A_1 = (0)$ or $B_1 = (0)$ or $C_1 = (0)$. If $A_1 = (0)$, then the deletion of the vertex corresponding to this trivial component leaves the order of the tripartite tournament to be $(p-1) \times q \times r$. Thus these are $t(p-1,q,r)$ tripartite score lists of order $p \times q \times r$ such that $A_1 = (0)$. Similarly there are $t(p,q-1,r)$ and $t(p,q,r-1)$ tripartite score lists of order $p \times q \times r$ such that $B_1 = (0)$ and $C_1 = (0)$ respectively. Hence $t_o(p,q,r) = t(p-1,q,r) + t(p,q-1,r) + t(p,q,r-1)$.||

5.2 TRIPARTITE TOURNAMENTS AND BIPARTITE TOURNAMENTS.

Let T be a tripartite tournament with partite sets X, Y and Z. In this section, we plan to study the relation between bipartite tournaments and tripartite tournaments. First we shall show that a tripartite tournament is a combination of three bipartite tournaments. Let T_1 be a bipartite tournament with partite sets X and Y and let $(u,v) \in E(T_1)$ iff $(u,v) \in E(T)$. Let T_2 be a bipartite tournament with partite sets Y and Z such that $(v,w) \in E(T_2)$ iff $(v,w) \in E(T)$.

Let T_3 be a bipartite tournament with partite sets X and with $(w,u) \varepsilon E(T_3)$ iff $(w,u) \varepsilon E(T)$. Thus we observe that a tripartite tournament T is a combination of three bipartite tournaments T_1, T_2 and T_3 such that $E(T) = E(T_1) \cup E(T_2) \cup E(T_3)$.

Let $X = \{u_1, u_2, \ldots, u_p\}$, $Y = \{v_1, v_2, \ldots, v_q\}$ and $Z = \{w_1, w_2, \ldots, w_r\}$ be the partite sets of a tripartite tournament T. To each vertex $u_i \varepsilon X$, we assign a vector $S(u_i)$ of dimension 3 as follows.

$S(u_i) = (0, a_{i2}, a_{i3})$ where

$a_{i2} = |\{v \varepsilon Y: (u_i, v) \varepsilon E(T)\}|$ and $a_{i3} = |\{w \varepsilon Z: (u_i, w) \varepsilon E(T)\}|$

Similarly to each vertex $v_j \varepsilon Y$ and $w_k \varepsilon Z$ we assign vectors $S(v_j)$ and $S(w_k)$ as follows.

$S(v_j) = (b_{j1}, 0, b_{j3})$ where $b_{j1} = |\{u \varepsilon X: (v_j, u) \varepsilon E(T)\}|$ and $b_{j3} = |\{w \varepsilon Z: (v_j, w) \varepsilon E(T)\}|$.

and

$S(w_k) = (c_{k1}, c_{k2}, 0)$ where $c_{k1} = |\{u \varepsilon X: (w_k, u) \varepsilon E(T)\}|$ and $c_{k2} = |\{v \varepsilon Y: (w_k, v) \varepsilon E(T)\}|$.

Let

$$\vec{A} = ((0, a_{12}, a_{13}), (0, a_{22}, a_{23}), \ldots, (0, a_{p2}, a_{p3}))$$

$$\vec{B} = ((b_{11}, 0, b_{13}), (b_{21}, 0, b_{23}), \ldots, (b_{q1}, 0, b_{q3})) \qquad (5.2.1)$$

and

$$\vec{C} = ((c_{11}, c_{12}, 0), (c_{21}, c_{22}, 0), \ldots, (c_{r1}, c_{r2}, 0)).$$

We call $\vec{A}$, $\vec{B}$ and $\vec{C}$ as the score vectors of a tripartite tournament T.

Also let $\vec{A}' = ((0,q-a_{p2}, r-a_{p3}),\ldots,(0,q-a_{22},r-a_{22}),$ $(0,q-a_{12},r-a_{13}),$ $\vec{B}' = ((p-b_{q1},0,r-bq_3),\ldots,(p-b_{21},0,r-b_{23}),$ $(p-b_{11},0,r-b_{13}))$ and $\vec{C}' = ((p-c_{r1},q-c_{r2},0),\ldots,(p-c_{21},q-c_{22},0),$ $(p-c_{11},q-c_{12},0))$. Then the following are equivalent.

$\vec{A}$, $\vec{B}$ and $\vec{C}$ are the score vectors of some tripartite tournament.

$\vec{A}'$, $\vec{B}'$ and $\vec{C}'$ are the score vectors of some tripartite tournament.

The score vectors $\vec{A}'$, $\vec{B}'$ and $\vec{C}'$ are said to be the duals of the tripartite score vectors $\vec{A}$, $\vec{B}$ and $\vec{C}$ respectively.

If $A = (a_1,a_2,\ldots,a_p)$, $B = (b_1,b_2,\ldots,b_q)$ and $C = (c_1,c_2,\ldots,c_r)$ are the score lists of some tripartite tournament T, then

$a_i = a_{i2} + a_{i3}$, $1 \leq i \leq p$; $b_j = b_{j1} + b_{j3}$, $1 \leq j \leq q$ and $C_k = c_{k1} + c_{k2}$, $1 \leq k \leq r$.

Also if $A_1 = (a_{12},a_{22},\ldots,a_{p2})$, $B_1 = (b_{11},b_{21},\ldots,b_{q1})$ $A_2 = (b_{13},b_{23},\ldots,b_{q3})$, $B_2 = (c_{12},c_{22},\ldots,c_{r2})$, $A_3 = (c_{11},c_{21},\ldots,c_{r1})$ and $B_3 = (a_{13},a_{23},\ldots,a_{p3})$, then A_i and B_i are the score lists of bipartite tournaments T_i for $1 \leq i \leq 3$.

Thus by Theorem 4.1.1, we get the following result.

<u>Theorem 5.2.1</u>. Let $\vec{A} =((0,a_{12},a_{13}), (0,a_{22},a_{23}),\ldots,(0,a_{p2},a_{p3}))$, $\vec{B} = ((b_{11},0,b_{13}), (b_{21},0,b_{23}),\ldots,(b_{q1},0,b_{q3})$ and $\vec{C} = ((c_{11},c_{12},0), (c_{21},c_{22},0),\ldots,(c_{r1},c_{r2},0))$ be the sets of

vectors with entries being nonnegative integers. Let $\vec{A}_1$ be obtained from $\vec{A}$ by deleting the ith entry i.e. $(0,a_{12},a_{i3})$ and $\vec{B}_1$ and $\vec{C}_1$ be obtained from $\vec{B}$ and $\vec{C}$ by reducing each of the largest $q-a_{12}$ remaining entries from b_{j1}'s by 1 and each of the largest $r-a_{i3}$ remaining entries from c_{k1}'s by 1 respectively. Then $\vec{A}$, $\vec{B}$ and $\vec{C}$ are the score vectors of some tripartite tournament iff $\vec{A}_1$, $\vec{B}_1$ and $\vec{C}_1$ are.

<u>Proof</u>. It follows from Theorem 4.1.1.||

We characterise tripartite score vectors below. This result is analogous to theorem 4.1.2. The proof being similar to the proof of Theorem 4.1.2 we omit it here.

<u>Theorem 5.2.2</u>. Let $\vec{A} = ((0,a_{12},a_{13}), (0,a_{22},a_{23}),\ldots,(0,a_{p2},a_{p3}))$, $\vec{B} = ((b_{11},0,b_{13}), (b_{21},0,b_{23}),\ldots,(b_{q1},0,bq_3))$ and $\vec{C} = ((c_{11},c_{12},0), (c_{21},c_{22},0),\ldots,(c_{r1},c_{r2},0))$ be three sets of vectors with entries being nonnegative integers such that

$a_{12} \leq a_{22} \leq\ldots\leq a_{p2}$; $a_{13} \leq a_{23}\leq\ldots\leq a_{r3}$;
$b_{11} \leq b_{21} \leq\ldots\leq b_{q1}$; $b_{13} \leq b_{23}\leq\ldots\leq b_{q3}$;
$c_{11} \leq c_{21} \leq\ldots\leq c_{r1}$ and $c_{12} \leq c_{22} \leq\ldots\geq c_{r2}$.

Then $\vec{A}$, $\vec{B}$ and $\vec{C}$ are the score vectors of some tripartite tournament iff

$$\sum_{i=1}^{l} a_{i2} + \sum_{j=1}^{m} b_{j1} \geq lm \tag{5.2.2}$$

for $1 \leq l \leq p$, $1 \leq m \leq q$ where equality holds for $l = p$ and $m = q$.

$$\sum_{j=1}^{m} b_{j3} + \sum_{k=1}^{n} c_{k2} \geq mn \tag{5.2.3}$$

for $1 \leq m \leq q$ and $1 \leq n \leq r$ with equality holding for $m = q$ and $n = r$

and

$$\sum_{k=1}^{n} c_{k1} + \sum_{i=1}^{l} a_{i3} \geq nl \tag{5.2.4}$$

for $1 \leq n \leq r$ and $1 \leq l \leq p$ and the equality holds for $n = r$ and $l = p$.

While characterising the tripartite score lists in Theorem 5.1.2, we had three conditions. Now we show that all those three conditions can be obtained from Theorem 5.?.2.

First we show that condition (1) of Theorem 5.1.2 is satisfied. From equations (5.2.2), (5.2.3) and (5.2.4), we get

$$\sum_{i=1}^{l} a_i + \sum_{j=1}^{m} b_j + \sum_{k=1}^{n} c_k = \left(\sum_{i=1}^{l} a_{i2} + \sum_{j=1}^{m} b_{j1} \right)$$

$$\left(\sum_{j=1}^{m} b_{j3} + \sum_{k=1}^{n} c_{k2} \right) + \left(\sum_{k=1}^{n} c_{k1} + \sum_{i=1}^{l} a_{i3} \right)$$

$\geq lm + mn + nl$ for $1 \leq l \leq p$, $1 \leq m \leq q$ and $1 \leq n \leq r$.

We note from equations (5.2.2), (5.2.3) and (5.2.4) that equality holds for $l=p$, $m=q$ and $n=r$. Thus condition (1) of Theorem 5.1.2 is satisfied.

Now we turn to condition (2). Let $a_1 = a_2 = \ldots = a_o = 0$. Since $a_i = a_{i2} + a_{i3}$, hence $a_{i2} = a_{i3} = 0$ for $1 \leq i \leq o$.

Substituting $l = 0$ and $m = 1$ in equation (5.2.2), we get $b_{11} \geq 0$ implying $b_1 \geq 0$ as $b_1 = b_{11} + b_{13}$. Similarly we can show that $c_1 \geq 0$. Thus condition (2) of Theorem 5.1.2 is satisfied. Similarly we can show that condition (3) of Theorem 5.1 is satisfied but for this case we make use of the duals $\vec{A}'$, $\vec{B}'$ and $\vec{C}'$.

We have the following characterisation of bipartite score lists, See [92].

__Theorem 5.2.3.__[92] . Let $A = (a_1, a_2, \ldots, a_m)$ and $B = (b_1, b_2, \ldots, b_n)$ be the lists of nonnegative integers with A in a nonincreasing order. Then A and B are the score lists of some bipartite tournament iff

$$\sum_{i=1}^{k} a_i \leq \sum_{j=1}^{n} \min(k, m-b_j) \quad \text{for } 1 \leq k \leq m \qquad (5.2.5)$$

with equality holding for $k = m$.

Furthermore, the bipartite tournament is strong iff the inequality is strict for all $k < m$ and $0 < b_j < m$ for all $j \leq n$.

An analogous result in the case of tripartite tournaments is as follows.

__Theorem 5.2.4.__ Let $\vec{A} = ((0, a_{12}, a_{13}), (0, a_{22}, a_{23}), \ldots, (0, a_{p2}, a_{p3}))$ with $a_{12} \geq a_{22} \geq \ldots \geq a_{p2}$, $\vec{B} = ((b_{11}, 0, b_{13}), (b_{21}, 0, b_{23}), \ldots, (b_{q1}, 0, b_{q3}))$ with $b_{13} \geq b_{23} \geq \ldots \geq b_{q3}$ and $\vec{C} = ((c_{11}, c_{12}, 0), (c_{21}, c_{22}, 0), \ldots, (c_{r1}, c_{r2}, 0))$ with $c_{11} \geq c_{21} \geq \ldots \geq c_{r1}$,

with the entries being nonnegative integers. Then $\vec{A}$, $\vec{B}$ and $\vec{C}$ are the score vectors of some tripartite tournament iff

$$\sum_{i=1}^{l} a_{i2} \leq \sum_{j=1}^{q} \min(l, p-b_{j1}) \quad \text{for} \quad 1 \leq l \leq p \qquad (5.2.6)$$

equality holding for $l = p$

$$\sum_{j=1}^{m} b_{j3} \leq \sum_{k=1}^{r} \min(m, q-c_{k2}) \quad \text{for} \quad 1 \leq m \leq q \qquad (5.2.7)$$

with equality sign for $m = q$

and

$$\sum_{k=1}^{n} c_{k1} \leq \sum_{i=1}^{p} \min(n, r-a_{i3}) \quad \text{for} \quad 1 \leq n \leq r$$

with equality holding for $n = r$.

We now discuss the strongness of the score lists. Harary and Moser [54] have characterised the strong score sequence of ordinary tournaments.

<u>Theorem 5.2.5 [54]</u> . Let $S = (s_1, s_2, \ldots, s_n)$ be a score sequence of an n-tournament. If each score s_i satisfies $\frac{1}{4}(n-1) \leq s_i \leq \frac{3}{4}(n-1)$, then the tournament is strong.

The following analogue has been obtained by Beinke and Moon [10] (This result was earlier given by Moon [80]) in the case of bipartite tournaments.

<u>Theorem 5.2.6 [80]</u> . Let T be a bipartite tournament with

the score lists $A = (a_1, a_2, \ldots, a_m)$ and $B = (b_1, b_2, \ldots, b_n)$ satisfying $\frac{1}{4} n < a_i < \frac{3}{4} n$ for $1 \leq i \leq m$ and $\frac{1}{4} m < b_j < \frac{3}{4} m$ for $1 \leq j \leq n$. Then T is strong.

As an extension of this result, we obtain the following characterisation of strong tripartite tournaments in terms of the score vectors. The proof is similar to that of Theorem 5.2.6 and hence we omit it.

Theorem 5.2.7. Let T be a tripartite tournament with score vectors $\vec{A} = ((0, a_{12}, a_{13}), (0, a_{22}, a_{23}), \ldots, (0, a_{p2}, a_{p3}))$, $\vec{B} = ((b_{11}, 0, b_{13}), (b_{21}, 0, b_{23}), \ldots, (b_{q1}, 0, b_{q3}))$ and $\vec{C} = (c_{11}, c_{12}, 0), (c_{21}, c_{22}, 0), \ldots, (c_{r1}, c_{r2}, 0))$ satisfying

(I) $\frac{1}{4} q < a_{i2} < \frac{3}{4} q$ for $1 \leq i \leq p$ and

$$\frac{1}{4} p < b_{j1} < \frac{3}{4} p \quad \text{for } 1 \leq j \leq q \tag{5.2.9}$$

(II) $\frac{1}{4} r < b_{j3} < \frac{3}{4} r$ for $1 \leq j \leq q$ and $\frac{1}{4} q < a_{22} < \frac{3}{4} q$

$$\text{for } 1 \leq k \leq r \tag{5.2.10}$$

(III) $\frac{1}{4} p < c_{k1} < \frac{3}{4} p$ for $1 \leq k \leq r$ and $\frac{1}{4} r < a_{i3} < \frac{3}{4} r$

$$\text{for } 1 \leq i \leq p \tag{5.2.11}$$

Then T is strong.

We now characterise strong tripartite tournaments interms of their score lists.

Theorem 5.2.8. Let T be a tripartite tournament with the score lists $A = (a_1, a_2, \ldots, a_p)$, $B = (b_1, b_2, \ldots, b_q)$ and

$C = (c_1, c_2, \ldots, c_r)$ satisfying

$$\frac{1}{4}(q+r) < a_i < \frac{3}{4}(q+r) \quad \text{for } 1 \leq i \leq p \qquad (5.2.12)$$

$$\frac{1}{4}(r+p) < b_j < \frac{3}{4}(r+p) \quad \text{for } 1 \leq j \leq q \qquad (5.2.13)$$

and

$$\frac{1}{4}(p+q) < c_k < \frac{3}{4}(p+q) \quad \text{for } 1 \leq k \leq r \qquad (5.2.14)$$

Then T is strong.

Proof. This follows from the Theorem 5.2.6.↓↓.

We know that an ordinary tournament can be obtained from any other having the same scores by a sequence of arc reversal of 3-cycles (see [92, 93]). Beineke and Moon [10] have given an analogous result in the case of bipartite tournaments. The result is, ''if the two bipartite tournaments have the same score lists, then each can be transformed in to the other by successively reversing the arcs of 4-cycles.'' We present here a conjecture in the case of tripartite tournaments.

Conjecture 5.2.1. If the two tripartite tournaments have the same score lists, then each can be transformed into the other by successively reversing the arcs of 3-cycles.

If the conjecture can be proved to be true, then in each kind of tournaments, two tournaments having the same score lists can be transformed in to the other by successively reversing the arcs of cycles of minimum length.

Now, we shall study the existence of cycles in each kinds of tournaments. Harary and Moser [54] have established that every strong n-tournament has cycles of length $3, 4, \ldots, n$. But this is not the case in bipartite and tripartite tournaments, where as the strongness certainly guarentees the existence of some cycles. Harary and Moser [54] have also shown that if an ordinary tournament has an r-cycle, then it has cycles of lengths $3, 4, \ldots, r$. We know that no cycle in a bipartite tournament can have odd length. First, we define a F_{4r} bipartite tournament and then study the existence of cycles in the case of bipartite tournaments.

Definition 5.2.1. A bipartite tournament F_{4r} has partite sets $X = \{v_1, v_3, \ldots, v_{4r-1}\}$ and $Y = \{v_2, v_4, \ldots, v_{4r}\}$. The arc set is such that (v_i, v_j) is an arc if $j-i \equiv 1 (\text{mod } 4)$. We note that all cycles of F_{4r} are multiples of 4 and that all multiples of 4 (upto the rth) occur. Beineke and Little [11] have shown the following.

Theorem 5.2.9. If a bipartite tournament has a cycle of length $2n$, then it has directed cycles of all smaller even lengths unless n is even and the 2n-cycle induces one special digraph, namely F_{4r} where $r = n/2$.

Now, we shall study the existence of cycles in tripartite tournaments. First we give a definition.

<u>Definition 5.2.2.</u> We define a tripartite tournament F_{3r} as follows. Let $X = \{v_1, v_4, \ldots, v_{3r-2}\}$, $Y = \{v_2, v_5, \ldots, v_{3r-1}\}$ and $Z = \{v_3, v_6, \ldots, v_{3r}\}$. The arc set is such that (v_i, v_j) is an arc if $j-i \equiv 1 \pmod 3$. We note that all the cycles of F_{3r} are multiples of 3 and all multiples of 3 (upto the rth) exist. In this case we note that $A=B=C=(r, r, \ldots, r)$, where A, B and C are the score lists of F_{3r}.

The result that we obtain is as follows.

<u>Theorem 5.2.10.</u> Let T be a tripartite tournament with the partite sets X, Y and Z. Let the score of each vertex of some partite set be zero. Then the existence of c_{2n} implies the existence of c_{2s} for $2 \leq s \leq n$, unless n is even and the 2n-cycle induces one special digraph, namely F_{4r}, where $r = n/2$.

<u>Proof.</u> Let $s(w) = 0$ for every vertex w in Z. Thus no cycle of T can pass through the vertices of Z. Hence any cycle of T has to pass through the only vertices of X and Y. But the digraph induced by the vertices of X and Y is a bipartite tournament. Now the result follows from Theorem 5.2.9.||

We now report a conjecture which deals with the existence of cycles in tripartite tournaments.

<u>Conjecture 5.2.2.</u> In a tripartite tournament T, the existence of c_n implies the existence of c_k for $3 \leq k \leq n$, except

when the tripartite tournament induced by the vertices of C_n is isomorphic to F_{3r} where $r = n/3$ or T has the structure as described in Theorem 5.2.10.

We conclude this section with some open problems which follow .

(I) The characterisation of the simple tripartite score lists and

(II) The characterisation of self-converse tripartite score lists.

REFERENCES

1. A.V. Aho, J.E. Hopcroft and J.D. Ullman, The Design and Analysis of Computer Algorithms, Addison-Wesley, Reading, MA (1974).

2. B. Alspach, Cycles of each lengths in regular tournaments, Canad. Math. Bull. 10 (1967) 283-286.

3. Tom M. Apostol, Introduction to Analytic Number Theory, Springer-Verlag, New York (1976).

4. Peter Avery, Condition for a tournament score sequence to be simple, J. Graph Theory, 4 (1980) 157-164.

5. L. Babai, On the isomorphism problem, Proc. FCT Conf., Poznan-Kornak (1977).

6. L. Babai, L. Kucera, Graph canonization with linear expected time, preprint, (1978).

7. L. Babai, P. Erdős and S.M. Selkow, Random graph isomorphism, SIAM J. Comput., Vol. 9, No. 3 (1980) 628-635.

8. M. Behzad, G. Chartrand, and L. Lesniak-Foster, Graphs and Digraphs, Prindle, Weber and Schmidt, Boston (1979).

9. L.W. Beineke, A tour through tournaments or bipartite and ordinary tournaments : a comparative survey, British Combinatorial Conference, 8th, Swansea (1981) Combinatories, Proceedings, (Ed. H.N.V. Temperley), Cambridge University Press, 41-45.

10. L.W. Beineke and J.W. Moon, On bipartite tournaments and scores, Proc. of the Fourth International Conference on the Theory and Applications of Graphs, Kalmazoo, MI, May 6-9 (1980) 55-71.

11. L.W. Beineke and C.H.C. Little, Cycles in bipartite tournaments, J. Combinatorial Theory, Series B, 32(1982) 140-145.

12. A.T. Berztiss, Data Structure Theory and Practice, Academic Press (1971).

13. A.T. Berztiss, A backtrack procedure for isomorphism of directed graphs, J. ACM, 20 (1973) 365-377.

14. B. Bollabas, O. Frank and M. Karonski, On 4-cycles in random bipartite tournaments, J. of Graph Theory, Vol. 7 (1983) 183-194.

15. J.A. Bondy and U.S.R. Murty, Graph Theory with Applications, Macmillan, London (1976).

16. K.S. Booth, Problems polynomially equivalent to graph isomorphism (abstract) in : Algorithms and Complexity (J.E. Traub, ed.) Academic Press (1976) 435.

17. K.S. Booth, Isomorphism testing for graphs, semigroups and finite automata are polynomially equivalent problems, SIAM J. of Comput. 7, 3(1978) 273-279.

18. K.S. Booth and G.S. Lueker, Linear algorithms to recognise interval graphs and test for consecutive ones property, Proc. of the 7th Annual ACM Symposium on the Theory of Computing (May 1975) 255-265.

19. K.S. Booth and C.J. Colbourn, Problems polynomially equivalent to graph isomorphism, TR-CS-77-04 (1979) Univ. of Waterloo.

20. P. Camion, Chemins et circuits hamiltoniens des graphes complets, C.R. Acad. Sci. Paris (A) 249 (1959) 2151-2152.

21. P. Camion, Quelques proprietes des chemins et circuits hamiltoniens dans la theorie des graphes, Cahiers Centre Etudes Recherche Oper. 2 (1960) 5-36.

22. V. Chavtal, Monochromatic paths in edge-colored graphs, J. Combinatorial Theory (B) 13 (1972) 69-70.

23. C.J. Colbourn,A bibliography of graph isomorphism problem, TR-123-78, Department of Computer Science, University of Toronto.

24. C.J. Colbourn, The Complexity of Graph Isomorphism and Related Problems, Ph.D. thesis, Univ. of Toronto (1980).

25. C.J. Colbourn, On testing isomorphism of permutation graphs, Networks, Vol. 11 (1981) 13-21.

26. C.J. Colbourn and M.J. Colbourn, Isomorphism problems involving self-complementary graphs and tournaments, Proc. of the eighth Manitoba Conference on Numerical Mathematics and Computing (1978) 153-164.

27. S.A. Cook, The complexity of theorem proving procedures, Proc. of the Third Annual ACM Symposium on the Theory of Computing (1971) 151-158.

28. D.G. Corneil, Graph Isomorphism, Ph.D. thesis, Univ. of Toronto (1968).

29. D.G. Corneil, An algorithm for determining the automorphism partition of an undirected graph, BIT 12 (1972) 161-171.

30. D.G. Corneil, The analysis of graph theoretical algorithms, Proc. of the Fifth Southeast Conference on Combinatorics, Graph Theory and Computing (1974) 3-38.

31. D.G. Corneil, Recent results on the graph isomorphism problem, Proc. of the Eighth Manitoba Conference on Numerical Mathematics and Computing (1978) 13-31.

32. D.G.Corneil and C.C. Gotlieb, An efficient algorithm for graph isomorphism, J. of the ACM 17, 1 (1970) 51-64.

33. D.G. Corneil and D.G. Kirkapatrick, A theoretical analysis of various heuristics for the graph isomorphism problem, SIAM J. Comput. Vol. 9, No. 2 (1980) 281-297.

34. R.L. Davis, Structures of dominance relations, Bull. Math. Biophys. 16 (1954) 131-140.

35. N. Deo, Graph Theory with Applications to Engineering and Computer Science, Prentice Hall (1974).

36. N. Deo, J.M. Davis and R.E. Lord, A new algorithm for digraph isomorphism, BIT 17, 1 (1977) 16-30.

37. R.J. Douglas, Tournaments that admit exactly one Hamiltonian circuit, Proc. London Math. Soc. (3) 3(1970) 716-730.

38. G.P. Egorycev, The method of coefficients and a generating function for the number of non-isomorphic tournaments that have a unique Hamiltonian contour, Kombinatornyi Anal. Vyp. 3 (1974) 5-9.

39. W. J. R. Eplett, Self-converse tournaments, Canad. Math. Bull. 22 (1979) 23-24.

40. Shimon Even, Graph Algorithms, Computer Science Press (1979).

41. T. Feng, Wu Zheng-Sheng and Z. Cun-Quan, Cycles of each length in tournaments, J. Comb. Theory (B) 33 (1982) 245-255.

42. R. Forcade, Parity of paths and circuits in tournaments, Discrete Math. 6 (1973) 115-118.

43. L. R. Ford and D.R. Fulkerson, Flows in Net-works, Princeton Univ. Press, Princeton (1962).

44. R.W. Frucht, Lattices with a given group of automorphisms, Canad. J. Maths. 2 (1950) 417-419.

45. T. Gallai, On directed paths and circuits, in Theory of Graphs (eds.P. Erdős and G. Katona) Academic Press, New York (1968) 115-118.

46. M.R. Garey, On enumerating tournaments that admit exactly one Hamiltonian circuit, J. Comb. Theory (B) 13 (1972) 266-269.

47. M.R. Garey, D.S. Johnson and L. Stockmeyer, Some simplified NP-complete graph problems, Theor. Comput. Sci. 1(1976) 237-267.

48. M.R. Garey and D.S. Johnson, Computers and Interactibility: A Guide to the Theory of NP-Completeness, Freeman, San Francisco (1979).

49. G. Gati, Further annotated bibliography on the isomorphism disease, J. Graph Theory 3 (1979) 95-109.

50. B. Grünbaum, Antidirected Hamiltonian paths in tournaments, J. Comb. Theory (B) 11 (1971) 249-257.

51. A Gyarafas and J. Lehel, A Ramsey-type problem in directed and bipartite graphs, Periodica Math. Hungar. 3 (1973) 299-304.

52. F. Harary, Graph Theory, Addison-Wesley, Reading, Mass. (1969).

53. F. Harary, R.Z. Norman and D. Cartwright, Structural Models : An Introduction to the Theory of Directed Graphs, John Wiley and Sons, New York (1965).

54. F. Harary and L. Moser, The theory of round robin tournaments, Amer. Math. Monthly 73 (1966) 231-246.

55. F. Harary and E.M. Palmer, Graphical Enumeration, Academic Press, New York (1973).

56. S. Hedetetniemi, Private communications (1977).

57. G. Heil, Private communications (1974).

58. J.E. Hopcroft and R.E. Tarjan, Isomorphism of planar graphs (working paper) in : Complexity of Computer Computations (eds. R.E. Miller and J.W. Thatcher) Plenum (1972) 131-152.

59. J.E. Hopcroft and J.K. Wong, Linear time algorithm for isomorphism of planar graphs, Proc. of the Sixth Annual ACM Symp. on the Theory of Computing (1974) 172-184.

60. O.S. Jakobsen, Cycles and paths in tournaments, Thesis, Univ. of Aarhus, Denmark (1972).

61. C.R. Johnson and F.T. Leighton, Efficient linear algebraic algorithm for determination of isomorphism in pairs of undirected graphs, J. Research of the National Bureau of Standards B 80,4 (1976) 447-483.

62. R.M. Karp, Reducibility among combinatorial problems in : Complexity of Computer Computations (eds. R.E. Miller and J.W. Thatcher) Plenum Press (1972) 85-104.

63. R.M. Karp, Probabilistic analysis of a canonical numbering algorithm for graphs, Proc. of Symposia in Pure Mathematics, 34 (1974) 365-378.

64. R.M. Karp, Recent advances in the probabilistic analysis of algorithms, Lecture Notes in Computer Science 71 (1979) 338-339.

65. M. Klawe, Private communications (1978).

66. A. Kotzig, The decomposition of a directed graph into quadratic factors consisting of cycles, Acta. Foc. Rerum Natur. Univ. Comenian Math. Publ. 22 (1969) 27-29.

67. D. Kozen, Complexity of finitely presented algebras, Proc. of the Ninth Annual ACM Symp. on the Theory of Computing (1977) 164-177.

68. D. Kozen, A clique problem equivalent to graph isomorphism, SIGACT News 10, 2 (1978) 50-52.

69. R.E. Ladner, The structure of polynomial reducibility, J. ACM 22 (1975) 155-171.

70. H.G. Landau, On dominance relations and the structure of animal societies, III : The condition for a score structure, Bull. Math. Biophysics, 15 (1953) 143-148.

86. R.C. Read and D.G. Corneil, The graph isomorphism disease, J. Graph Theory (1977) 339-363.

87. L. Redei, Ein kombinatorischer Satz, Acta Litt. Sci. Szeged 7 (1934) 39-43.

88. K.B. Reid and L.W. Beineke, Selected Topics in Graph Theory (eds.L.W. Beineke and R.J. Wilson) Academic Press (1979) 169-204.

89. F.S. Roberts, Discrete Mathematical Models, with applications to social, biological and environmental problems, Prentice-Hall, Inc., New Jersey (1976).

90. M. Rosenfeld, Antidirected Hamiltonian paths in tournaments, J. Comb. Theory (B) 12 (1972) 93-99.

91. M. Rosenfeld, Antidirected Hamiltonian circuits in tournaments, J.Comb. Theory (B) 16 (1974) 234-242.

92. H.J. Ryser, Combinatorial properties of matrices of zeros and ones, Canad. J. Math. 9 (1957) 371-377.

93. H.J. Ryser, Matrices of zeros and ones in combinatorial Mathematics, Recent Advances in Matrix Theory, Univ. of Wisconsin Press (Madison) (1964) 103-124.

94. D.C.Schmidt and L.E. Druffel, A fast backtracking algorithm to test directed graphs for isomorphism using distance matrices, J. ACM 23, 3 (1976) 433-445.

95. L. Stewart, Cographs-a class of tree representable graphs, M.Sc. thesis, Univ. of Toronto (1978).

96. C. Thomassen, Antidirected Hamiltonian circuits and paths in tournaments, Math. Ann. 201 (1973) 231-238.

97. C. Thomassen, Hamiltonian connected tournaments, preprint series, Aarhus Universitat 13 (1977) 38. Also in Journal of Combinatorial Theory (B) 28 (1980) 142-163.

98. S.H. Unger, GIT-a heuristic program for testing pairs of directed line graphs for isomorohism, Comm. ACM 7 (1964) 26-34.

99. B. Weisfeiler, On construction and identification of graphs, Lecture Notes in Mathematics 558 (1976).

100. B. Weisfeiler and A.A.Lehman, A reduction of graph to canonical form and an algebra arising during this construction, in Russian, Nauchno-Tech. Infor. Ser. 2, 9 (1968) 12-16.

101. H. Whitney, Congurent graphs and the connectivity of graphs, Amer. J. Math. 54 (1932) 150-168.

102. Z.S. Wu, K.M. Zhang and Y. Zhou, A necessary and sufficient condition for arc pancyclicity of tournaments, Sci. Sinica 8 (1981) 915-919.

103. Z.S. Wu, K.M. Zhang and Y. Zhou, A kind of counter example on arc-pancyclic tournaments, Acta Math. Appl. Sinica 6 (1983) No. 1.

104. Z.S. Wu, K.M. Zhang and Y. Zhou, Arc k-cyclic property of tournament T_{ss}, to appear.

105. C.Q. Zhang, Cycles of each length in a certain kind of tournament, Sci. Sinica 9 (1981) 1056-1062.

106. Ke-Min Zhang, Completely strong path-connected tournaments, J. Comb. Theory (B) 33 (1982) 166-177.

107. Y.J. Zhu, Study of tournaments - a survey, J. Qufu Teachers College, Special Issue (1980) 60-64.

108. Y.J. Zhu and F. Tian, On the strong path connectivity of a tournament, Sci. Sinica, Special Issue II (1979) 18-28.

A 00???

MATH-1983-D-UDD-SCO